



**FACHHOCHSCHULE
WIENER NEUSTADT**
Austrian Network for Higher Education

Prototyp einer Messeinrichtung zur Impedanzmessung am Handgelenk

Bachelorarbeit 1

Eingereicht von:

Mustafa Algan

Matrikelnummer:

51819221

am

**Fachhochschul-Bachelorstudiengang
Mechatronik/Mikrosystemtechnik**

Begutachter:

Dipl.-Ing. Dr. Koller Christian



Wiener Neustadt, 10. Jänner 2020

EIDESSTATTLICHE ERKLÄRUNG

Titel der Bachelorarbeit 1:

Prototyp einer Messeinrichtung zur Impedanzmessung am Handgelenk

Eingereicht von: Mustafa ALGAN

Matrikelnummer: 51819221

am: Fachhochschul Bachelorstudiengang

Mechatronik/Mikrosystemtechnik

Micro- and Nanoengineering

Begutachterin/Begutachter: Dipl.-Ing. Dr. Koller Christian

Wiener Neustadt am: 10. Jänner 2020

Ich erkläre an Eides statt, dass diese Arbeit ausschließlich von mir selbst verfasst wurde und ich diese Arbeit nicht zuvor an einer anderen Bildungseinrichtung zum Zwecke der Erlangung eines akademischen Grades vorgelegt habe.

Insbesondere wurden Beiträge anderer Personen entsprechend kenntlich gemacht sowie die in dieser Arbeit verwendeten Daten entsprechend der dargestellten Verfahren gewonnen und richtig wiedergegeben.

Wiener Neustadt, _____

Datum

Unterschrift

Kurzzusammenfassung:

In dieser Arbeit geht es um die Entwicklung einer Messeinrichtung zur Messung der elektrischen Impedanzen. Zu Beginn wird für die Messeinrichtung ein Schaltplan entworfen. Für die Messungen wurde ein Testaufbau gebaut. An dem Testaufbau sind acht Elektroden befestigt, mit dem 28 verschiedene Messungen möglich sind. Die Messdaten werden außerdem von der Messeinrichtung zu einem stationären Computer übertragen. Mit der fertigen Schaltung wurden die Messungen zum Schluss erfolgreich durchgeführt worden.

Schlagworte:

Elektrische Impedanz; Networkanalyzer; Arduino; AD5933

Abstract:

This work is about developing a measuring device which can measure electrical impedances. The measurement data are also transmitted from the measuring device to a stationary computer. The measurements were done on a test setup. Eight electrodes are attached to the test setup so that there are enough measuring points on the forearm. At the end the measurements were done with the finished circuit successfully. It planned to integrate the device in a wristband.

Keywords:

Electrical impedance; networkanalyzer; Arduino; AD5933

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Stand der Technik.....	2
1.1.1	Myo Armband	2
1.1.2	Dehnungsmessstreifen zur Gestenerkennung	2
1.1.3	Gestenerkennung mit Fotowiderständen	4
2	Theoretische Grundlagen	5
2.1	Impedanz	5
2.2	Ersatzmodell einer Zelle.....	7
2.3	I ² C Schnittstelle	8
2.4	Networkanalyzer	10
3	Technische Informationen.....	13
3.1	Messplatine.....	14
3.1.1	AD5933	14
3.1.2	Multiplexer	15
3.1.3	Status-LEDs	17
3.2	Phantom	17
3.3	Arduino Uno	18
3.3.1	I ² C Schnittstelle	19
3.3.2	VOUT Signal.....	19
4	Programmcode.....	21
5	Messung und Auswertung.....	23
5.1	Messaufbau	23

5.2	Kalibration der Messung.....	23
5.3	Betrag und Gain Factor Berechnungen	24
5.3.1	Betrag.....	24
5.3.2	Gain Factor.....	25
5.3.3	Impedanz.....	25
5.4	Das Verhalten der 28 Messwerte	26
5.4.1	Messung ohne Objekt.....	26
5.4.2	Messung mit einem Kunststoffkapsel.....	29
5.4.3	Messung mit Kunststoff und Metall Objekten	30
5.5	Vergleich der Kennlinien	32
6	Conclusio & Outlook.....	35
6.1	Conclusio	35
6.2	Outlook	36
	Abbildungsverzeichnis.....	37
	Tabellenverzeichnis.....	39
	Literaturverzeichnis.....	40
	Anhang	42

1 Einleitung

In unserer Gesellschaft ist die Technologie im Bereich der Sensoren sehr stark angestiegen und sie wird laufend immer weiter entwickelt. Eine Konsequenz davon ist es, dass Menschen leichter mit elektronischen Geräten interagieren können. Allein in Smartphones sind nahezu 14 Sensoren [1] eingebaut, die täglich von der Menschheit eingesetzt werden. Wie zum Beispiel der Bewegungssensor, der ermöglicht die Schritte des Benutzers zu zählen oder bei Spielen verschiedene Kommandos auszuführen. Doch in einigen Situationen ist es umständlicher Tasten zu bedienen, anstatt eine einfache Bewegung auszuführen. Die Befehlseingabe über Handgesten würde bei vielen Geräten eine weitaus komfortable Methode bieten, um mit Geräten schneller agieren zu können. Derzeit werden bereits Gestensteuerungen in Virtual Reality (VR) bzw. Augmented Reality (AR) Systemen implementiert [2]. Dies erlaubt den Benutzer ein besseres und reales Erlebnis zu erfahren. Derzeitige Systeme sind vor allem in der Lage Geschwindigkeitsänderungen und Verdrehungen des Handgelenks zu detektieren.

Gestenerkennung kann mittels verschiedener Methoden erfolgen. In dieser Arbeit wird das Prinzip der elektrischen Impedanz Tomographie (EIT) in Betracht gezogen. Die Inspiration für das Projekt kam von einer Veröffentlichung in dem eine entsprechende Gestenerkennung umgesetzt wurde. Es wurde ein tragbares System realisiert, mit dem eine EIT am Handgelenk durchgeführt werden könnte. Diese Impedanzmessung erfolgt über acht Elektroden.[3]

Zur Durchführung einer EIT am Handgelenk müssen vorher zwei Unteraufgaben bearbeitet werden.

1. Messschaltung und die Durchführung der Messungen
2. Darstellung einer Tomographie aus den Daten der Messungen

In dieser Arbeit wird der erste Bereich bearbeitet. Das Ziel ist es ein Messsystem zu entwickeln mit dem man eine elektrische Impedanz messen kann. Die Messung soll an einem Versuchsobjekt durchgeführt werden, welches in der Medizin Phantom genannt wird. In diesem Phantom soll es möglich sein verschiedene Objekte aus

unterschiedlichen Materialien zu vermessen und durch Widerstandsänderung auf die Positionierung der einzelnen Objekte rückschließen.

1.1 Stand der Technik

Es existieren schon sehr viele Unternehmen bzw. Projekte, die sich auf Gestenerkennungstechnologien fokussieren. In diesem Kapitel werden verschiedene Herangehensweisen für die Gestenerkennung gezeigt.

1.1.1 Myo Armband

Wenn man im Netz nach Gestensteuerung sucht, wird man bestimmt auf das Myo Armband stoßen (siehe Abbildung 1). Das Myo Armband beruht auf der Elektromyographie (EMG), bei dem die elektrischen Signale von menschlichen Muskeln gemessen werden. Das Armband enthält acht EMG Sensoren für die Messung der Muskelaktivitäten und des Weiteren wird ein Gyroskop, ein Beschleunigungssensor und ein Magnetometer zur Ermittlung von Bewegungen in neun Achsen eingesetzt. Eine integrierte Bluetooth Schnittstelle erlaubt eine einfache Verbindung mit dem PC. Das Myo Armband wurde in einer Publikation verwendet um das Spiel „Schere, Stein und Papier“ spielen zu können [4]. Dabei wurde versucht mittels eines Algorithmus die jeweilige Handstellung zu detektieren. Die Wahrscheinlichkeit lag bei ca. 78%, welches gut genug für diese Anwendung war. Die Ergebnisse von der Wahrscheinlichkeit können jedoch schwanken, aufgrund des Arms des Benutzers oder aufgrund von Umwelteinflüssen.[4]



Abbildung 1: Myo Armband

1.1.2 Dehnungsmessstreifen zur Gestenerkennung

Eine weiters interessantere Methode zur Gestenerkennung nutzt Dehnungsmessstreifen. Zwischen zwei Stoffbändern wurden sieben dieser

Sensoren mit leitenden Fäden eingenäht (siehe Abbildung 2), um eine Verbindung entlang des Armbandes herzustellen. Die Sensoren bestehen aus thermoplastischen und leitenden Nanopartikeln, die zu kleinen Filamenten verarbeitet werden. Diese werden gleichmäßig im Armband verteilt, um eine große Fläche des Armgelenks abzudecken. Bei Deformation der Haut verändern sich die Widerstandsmesswerte. Die Messwerte werden vom Armband über USB an den PC übertragen. In der Publikation wird erwähnt, dass die Übertragung auch für eine drahtlose Übertragung ausgestattet werden kann. Möglich war es 16 Gesten zu erkennen (siehe Abbildung 3), mit einer Wahrscheinlichkeit zwischen 87% und 95%. Ein Vorteil dieses Armbands ist das Gewicht, welches durch die Verwendung von Stoff sehr gering wurde.[5]

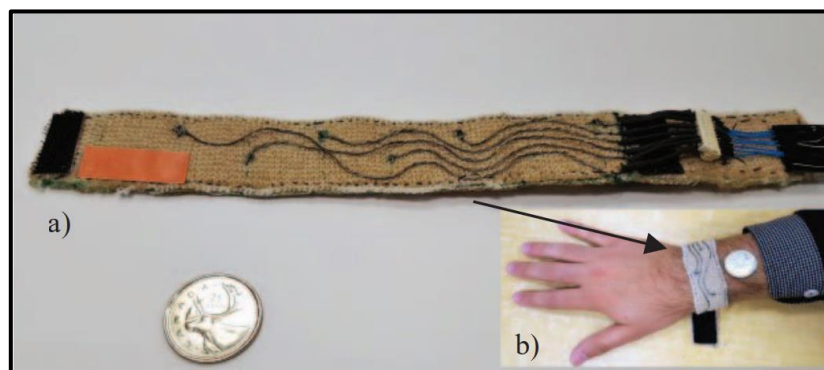


Abbildung 2: a) Ein Foto des tragbaren Systems mit Referenzmünze. b) Systems mit Details am Handgelenk getragen, und der gleichen Referenzmünze. [5]



Abbildung 3: 16 Gesten die mit dem Armband erkennbar sind. [5]

1.1.3 Gestenerkennung mit Fotowiderständen

Eine vollkommen andere Vorgehensweise für Gestenerkennung ist die Verwendung von Fotowiderständen. Bei Fotowiderständen ändert sich der Widerstand je nach eingebrachter Lichtintensität. In einer Publikation [6] wird beschrieben, dass eine 8x8 Matrix mit Fotowiderständen benutzt worden ist um den Schatten einer Hand zu erkennen und somit auch die erzeugte Geste. Der Schatten kommt zustande in dem eine Lichtquelle die Hand beim Durchführen der Geste beleuchtet (siehe Abbildung 4). Durch den Schatten wird ein zweidimensionales Abbild der Hand erkannt und durch die Lichtintensität kann auf eine drei dimensionale Geste schlussgefolgert werden. Mit dieser Methode konnte man 20 Gesten mit einer Wahrscheinlichkeit von über 90% erkennen. Der Vorteil dieser Methodik ist die Echtzeiterkennung, aber schlechte Lichtverhältnisse können sich auf die Erkennungswahrscheinlichkeit negativ auswirken. Außerdem ist man aufgrund der Fläche, in dem die Hand platziert wird begrenzt. Darüber hinaus ist der Aufbau stationär und nicht für mobile Anwendungen geeignet.[6]

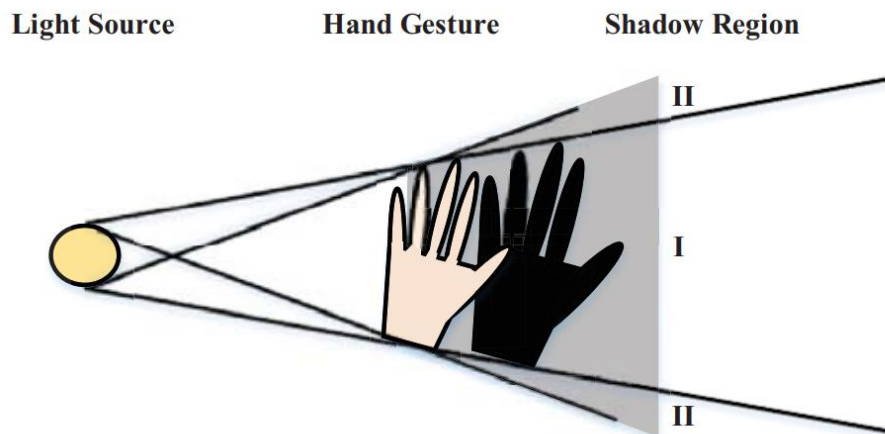


Abbildung 4: Lichtquelle mit dem Schatten. [6]

2 Theoretische Grundlagen

Im Folgenden werden die Grundlagen des Projektes näher erläutert. Zu Beginn wird das Konzept der Impedanz erklärt und anschließend ein Modell für die Impedanz einer Zelle. Weiters wurde im Projekt eine I²C Schnittstelle verwendet, weshalb die Schnittstelle etwas näher beschrieben wird. Zum Schluss wird der verwendete Networkanalyzer AD5933 verdeutlicht.

2.1 Impedanz

Impedanzen, auch Scheinwiderstand genannt, treten in der Wechselstromtechnik als ohmsche, induktive und kapazitive Widerstände meistens gemeinsam in Erscheinung. Aus dem ohmschen Gesetz ergibt sich die Impedanz $\underline{Z}$ aus der Spannung $\underline{U}$ und Strom $\underline{I}$ (siehe Abbildung 5) mit den jeweiligen Phasenverschiebungen φ_I & φ_U :

$$\underline{I} = I \angle \varphi_I \quad \underline{U} = U \angle \varphi_U$$
$$\underline{Z} = \frac{\underline{U}}{\underline{I}} = \frac{U \angle \varphi_U}{I \angle \varphi_I} = \frac{U}{I} \angle (\varphi_U - \varphi_I) = Z \angle \varphi$$

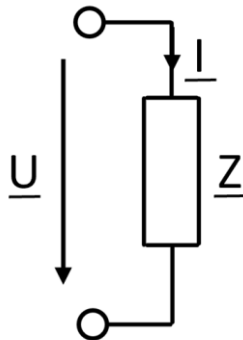


Abbildung 5: Schaltzeichen der Impedanz mit der anliegenden Spannung und der durchfließenden Strom

Die Impedanz $\underline{Z}$ besteht aus einem positiven Realteil (Resistanz R) und einem positiven bzw. negativen Imaginärteil (Reaktanz X):

$$Z = R + j \cdot X = Z \angle \varphi$$

An dem Realteil R gibt es keine Phasenverschiebung φ . Im Gegensatz dazu hat der Imaginärteil im idealem Fall eine Phasenverschiebung um 90° . Diese Verschiebung kann positiv sein, wenn der Strom der Spannung nacheilt, oder negativ, wenn die Spannung dem Strom nacheilt. Der Imaginärteil kann je nach Situation frequenzabhängig sein. Die Impedanz $\underline{Z}$ kann für das bessere Verständnis als Zeiger in der Gauß'schen Ebene dargestellt werden (siehe Abbildung 6 & 7). Der Stromzeiger bzw. Spannungszeiger wird mit φ_I bzw. φ_U angenommen. Das Argument der Impedanz $\varphi = \varphi_U - \varphi_I$ zeigt den Phasenwinkel zwischen dem Stromzeiger und Spannungszeiger. Die Werte befinden sich im Bereich zwischen $-90^\circ \leq \varphi \leq +90^\circ$. Die Impedanz verhält sich ohmsch-induktiv, wenn der Winkel im Bereich $0^\circ \leq \varphi \leq 90^\circ$ befindet, jedoch eine Impedanz mit dem Winkel $-90^\circ \leq \varphi \leq 0^\circ$ verhält sich ohmsch-kapazitiv.[7]

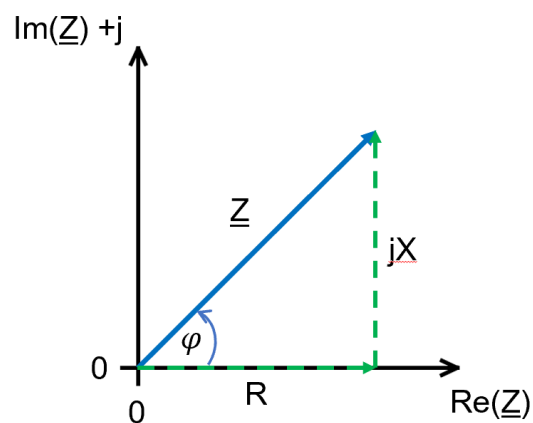


Abbildung 6: Ohmsch-induktive Impedanz

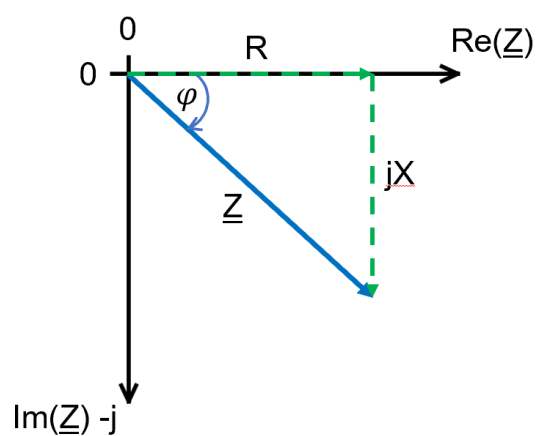


Abbildung 7: Ohmsch-kapazitive Impedanz

2.2 Ersatzmodell einer Zelle

Eine Zelle ist die grundlegendste strukturelle und funktionelle Einheit in lebenden Geweben. Ihr Struktur besteht aus intrazellulären und extrazellulären Materialien. Beide Materialien sind Elektrolyte und enthalten Ionen. Durch Ionen können elektrische Ladungen transportiert werden und somit elektrischer Strom geleitet werden. Die Impedanz des menschlichen Körpers kann so modelliert werden, als wäre sie aus resistiven und kapazitiven Elementen zusammengesetzt (siehe Abbildung 8). Um die elektrischen Eigenschaften lebendes Gewebes zu untersuchen, ist es von großer Hilfe, einen äquivalenten Stromkreis für die Zellen zu etablieren. Hier wirken Zellmembranen wie Kondensatoren und Körperflüssigkeiten (Intrazellulärflüssigkeit R_i & Extrazellulärflüssigkeit R_e) wie Widerstände.

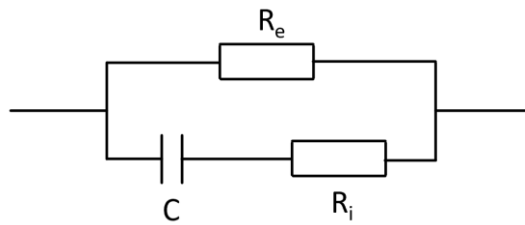


Abbildung 8: Ersatzschaltbild der Zelle

Die Zellmembranen verhalten sich wie Kondensatoren, wenn sie von Wechselstrom durchflossen werden. Bei niedrigen Frequenzen kann der Strom hauptsächlich durch extrazelluläre Flüssigkeiten fließen und bei hohen Frequenzen kann der Strom hauptsächlich durch Zellmembranen fließen. Mit der folgenden Herleitung kann die Frequenzabhängigkeit gesehen werden.

$$Z = R_e \parallel \left(R_i + \frac{1}{j\omega C} \right) = \frac{R_e (1 + j\omega C R_i)}{1 + j\omega C (R_e + R_i)}$$

Bei einer niedrigen Frequenz ω gegen 0 bleibt der Realteil, der extrazelluläre Widerstand R_e , übrig.

$$Z = \frac{R_e (1 + j\omega C R_i)}{1 + j\omega C (R_e + R_i)} = R_e$$

2.3 I²C Schnittstelle

Diese Schnittstelle wird auch Two Wire Interface (TWI) genannt und beruht auf dem Master-Slave-Bus Konzept, um mehrere ICs miteinander zu verknüpfen. Meistens wird diese Schnittstelle bei benachbarten Bauteilen verwendet, da die Leitung bei größerer Entfernung störanfälliger wird. Der Vorteil von I²C ist die Anzahl der Leitungen, da nur eine Leitung für die bidirektionale Datenübertragung (SDA=Serial Data) und eine weitere als Taktleitung (SCL=Serial Clock) benötigt wird. Mit nur zwei Leitungen ist die Schaltung somit sehr platzsparend umsetzbar.

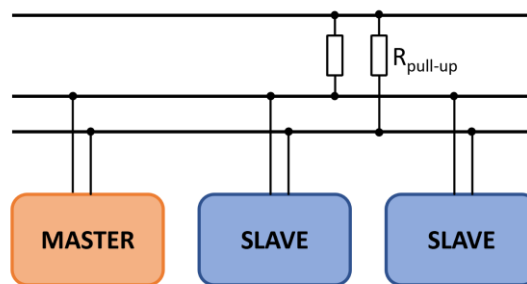


Abbildung 9: I²C Bus mit zwei Pull Up Widerständen, einem Master und zwei Slaves

Als zentrales Steuergerät (Master) wird in den meisten Fällen ein Microcontroller verwendet, mit dessen Hilfe man mehrere Peripheriebausteine (Slaves) ansteuern kann (siehe Abbildung 9). Die Auswahl eines Bausteins erfolgt über eine sieben Bit Adresse. Dabei sind die ersten vier Bits spezifisch vom Hersteller festgelegt und die restlichen drei Bits können, je nach Bauteil, über das Potential an einem weiteren Pin eingestellt werden. So ist es möglich eine Kommunikation mit mehreren Bausteinen auf der selben Leitung zu realisieren. Damit die SDA und SCL Leitungen einen definierten Zustand haben, müssen diese mit relativ hohen Pull Up Widerständen aktiv auf ein hohes Potenzial gesetzt werden.

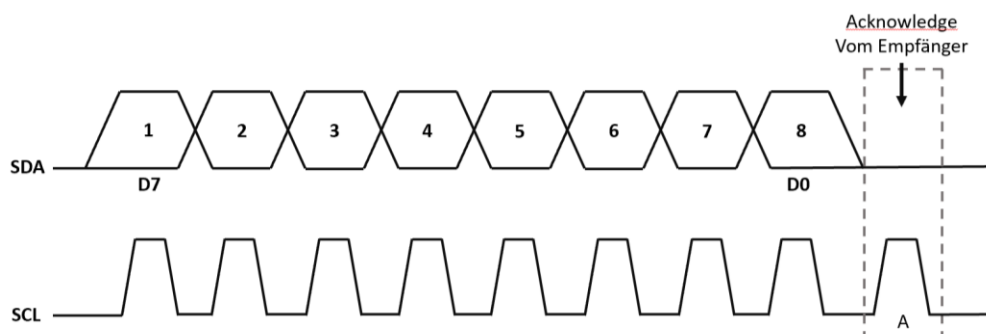


Abbildung 10: Zeitverhalten bei der Übertragung mit dem Acknowledge Bit

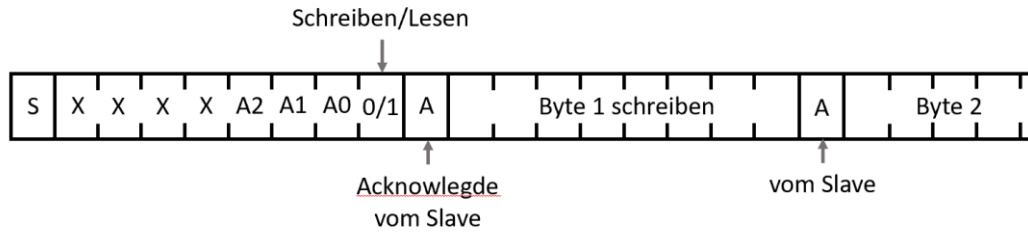


Abbildung 11: Aufbau des I²C Protokolls

Über die SDA Leitung werden die Daten wie bei einem Schieberegister bitweise vom Master an den Slave mit jedem Takt übertragen. Es gibt bestimmte Bedingungen die erfüllt sein müssen, sodass bei der Übertragung keine Probleme entstehen. Der Master muss ein Startbit senden mit dem die SDA eine Pegeländerung erfährt, wenn die SCL Leitung sich im High Zustand befindet. Danach wird die Adresse mit einem Datenrichtungsbit übertragen. Wenn der Master schreiben will, wird das Bit auf 0 gesetzt, andernfalls wird das Bit beim Lesen auf 1 gesetzt. Wenn der IC mit der richtigen Adresse angesprochen wurde, bestätigt der IC mit einem Acknowledge Bit (siehe Abbildung 10). Dann folgen die Datenpakete hintereinander, welche immer mit einem Acknowledge Bit bestätigt werden müssen (siehe Abbildung 11). Die Datenübertragung wird mit einem Stoppbit beendet (siehe Abbildung 12). Die Geschwindigkeit der Datenübertragung hängt von dem Mode ab (siehe Tabelle 1). Die verschiedenen Moden können über die Software festgelegt werden. [8]

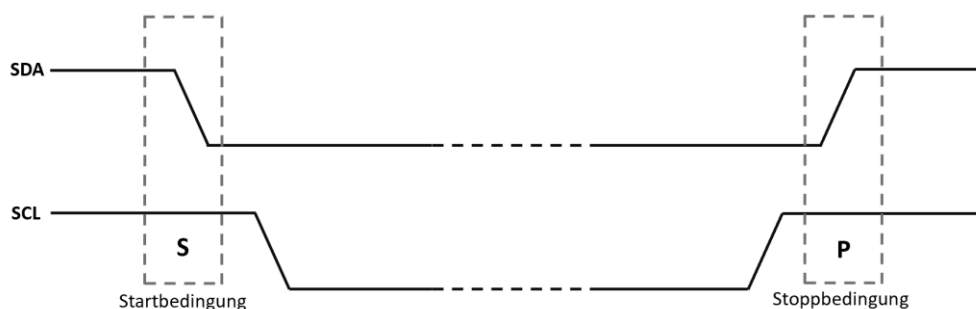


Abbildung 12: Start- und Stoppbedingung von der Datenübertragung

Mode	Maximale Übertragung	Richtung
Standard Mode	0,1 Mbit/s	bidirektional
Fast Mode	0,4 Mbit/s	
Fast Mode Plus	1,0 Mbit/s	
High Speed Mode	3,4 Mbit/s	
Ultra Fast Mode	5,0 Mbit/s	unidirektional

Tabelle 1: I²C Moden mit der Übertragungsgeschwindigkeiten

2.4 Networkanalyzer

Der Networkanalyzer AD5933 [9] ist ein hochpräzises Impedanzwandlersystem mit einem eingebauten Frequenzgenerator, dass auf einem einzelnen Chip untergebracht ist. Der interne Frequenzgenerator dient zur Bereitstellung des Signals zur Anregung der zu messenden Impedanz. Nach der Messung wird das Signal der Impedanz mit einem Verstärker verstärkt und mit Hilfe eines 12-Bit Analog Digital Konverters abgetastet. In der Schaltung befindet sich ein 1024 Bit DFT Prozessor, mit dem eine diskrete Fourier Transformation (DFT) des Signals durchgeführt wird. Man erhält hierdurch für jede gemessene Frequenz zwei Messwerte, eine Zahl für den Realteil und eine Zahl für den Imaginärteil. Mit diesen Werten kann man sich die Impedanz und das Argument der Phase errechnen. Mit dem AD5399 können Impedanzen zwischen $0,1\Omega$ bis $10M\Omega$ gemessen werden. Die Betriebsfrequenzen liegen zwischen 3kHz und 100kHz. Ein Blockschaltbild des IC AD5399 ist in Abbildung 13 dargestellt.

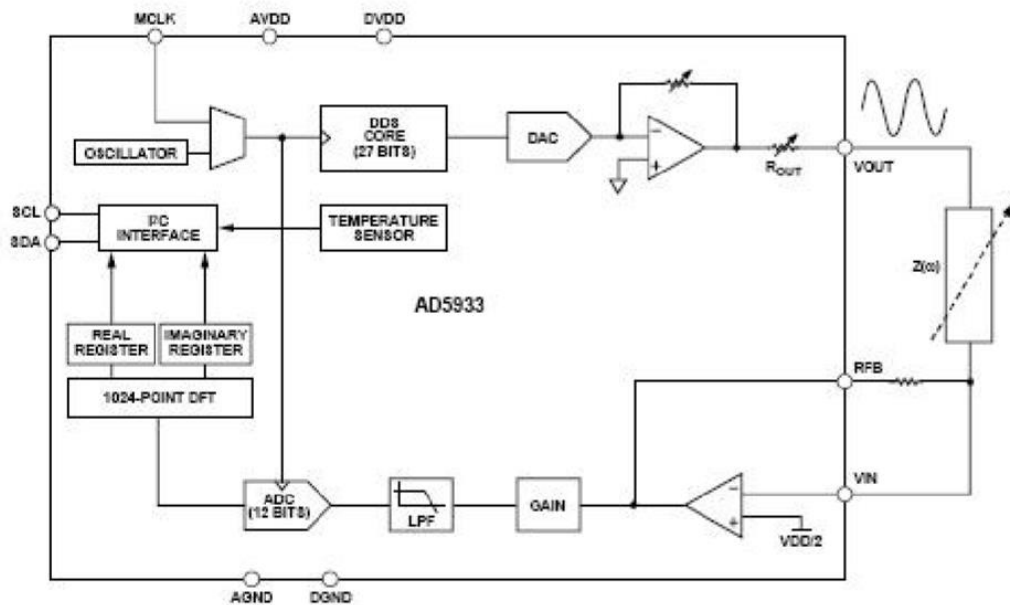


Abbildung 13: Blockschaltbild vom AD5933 [9]

In der Abbildung 13 befinden sich extern die Komponenten Z und R_{FB} . Z stellt die unbekannte Impedanz dar, welche gemessen werden soll. Der Feedback Widerstand R_{FB} bestimmt den Messbereich, denn die Linearität des Networkanalyzers befindet sich im Bereich um den R_{FB} Wert des Widerstandes. Um den linearen Messbereich zu bestimmen wird ein Kalibrationswiderstand R_{cal} bemessen. Der Widerstandswert des R_{cal} wird von uns definiert und statt Z in der Abbildung 13 platziert. Im Kapitel 5.2 wird dies näher erläutert.

Die Funktion des AD5933 besteht aus folgenden Hauptteilen:

- Der Sender erzeugt ein Signal unter Verwendung einer Direct Digital Synthesis (DDS) mit einem externen Oszillator (MCLK) oder einem internen Referenzoszillator. Danach wandelt ein DAC das digitale Signal in ein sinusförmiges Signal, welches als Anregungssignal für das zu prüfende Objekt verwendet wird. Das Signal des DAC wird durch eine programmierbare Verstärkung gesendet, bei dem man bei der Peak to Peak Ausgangsspannung zwischen 2V, 1V, 0,4V und 0,2V auswählen kann. Es ist auch möglich einen „frequency sweep“ durchzuführen. Dafür muss man eine Startfrequenz, ein Frequenzinkrement und eine Anzahl der Inkremente angeben.

- Wenn das zu prüfende Objekt durch das ausgegebene Spannungssignal angeregt wird, stellt sich ein Ansprechstrom ein. Dieser Strom fließt zum Strom-Spannungsverstärker auf der Empfängerseite (VIN). Dieser Verstärker ist programmierbar (programmable gain amplifier=PGA) mit einer Verstärkung von 1x oder 5x. Der Rückkopplungswiderstand RFB und der Verstärkungseinstellwiderstand des PGAs bestimmt den Signalpegel am ADC Eingang. Der Signalpegel sollte sich im linearen Bereich des ADC gehalten werden, also zwischen 0V und VDD. Die digitalen Daten vom ADC werden in den digitalen Signalverarbeitungsteil des Systems eingespeist, in dem eine diskrete Fouriertransformation (DFT) mit 1024 Abtastwerten für jeden Frequenzpunkt

$$X(f) = \sum_{n=0}^{1023} (x(n)(\cos(n) - j\sin(n)))$$

erfolgt.

Die Ausgangsdaten enthalten die Information über den Realteil und Imaginärteil (=kartesische Darstellung) der Messung. Für die beiden Teilergebnisse gibt es je zwei 8 Bit Register an Speicher, die zusammen einen Wert ergeben. Die ersten 8 Bit stellen das Most Significant Bit (MSB) und die letzten 8 Bit das Least Significant Bit (LSB) dar. Die Werte können über die serielle Schnittstelle I²C vom Mikrokontroller gelesen werden.

Um den wahren Betrag der Impedanz zu errechnen, wird sowohl Realteil und Imaginärteil als auch der Gain Factor benötigt siehe Kapitel 5.3.2.[9]

3 Technische Informationen

Der Messaufbau wird in drei Komponenten eingeteilt (siehe Abbildung 14). Einer selbst entworfener Messschaltung (siehe Abschnitt 3.1), einem Phantom (siehe Abschnitt 3.2) und einem Arduino Board (siehe Abschnitt 3.3). Im Mittelpunkt steht die Hauptkomponente, eine selbst entworfene Platine auf dem sich der Networkanalyzer AD5933 mit den zwei Multiplexer befindet. Bei dem Entwurf der Platine wurden die Dimensionen eines Arduino Uno Boards berücksichtigt. Die Platine wird direkt mit dem Arduino Board zusammengeführt, um durch Kabel entstehende Störungen zu vermeiden. Das Arduino Board stellt auch die Kommunikation zwischen Networkanalyzer und PC dar. Die Verbindung zwischen der Messplatine und dem Phantom wird über einem Flachbandkabel mittels eines Breakout Boards (siehe Abbildung 23) bewerkstelligt. Das Breakout Board dient zur einfachen Montage der Elektrodenkabel.

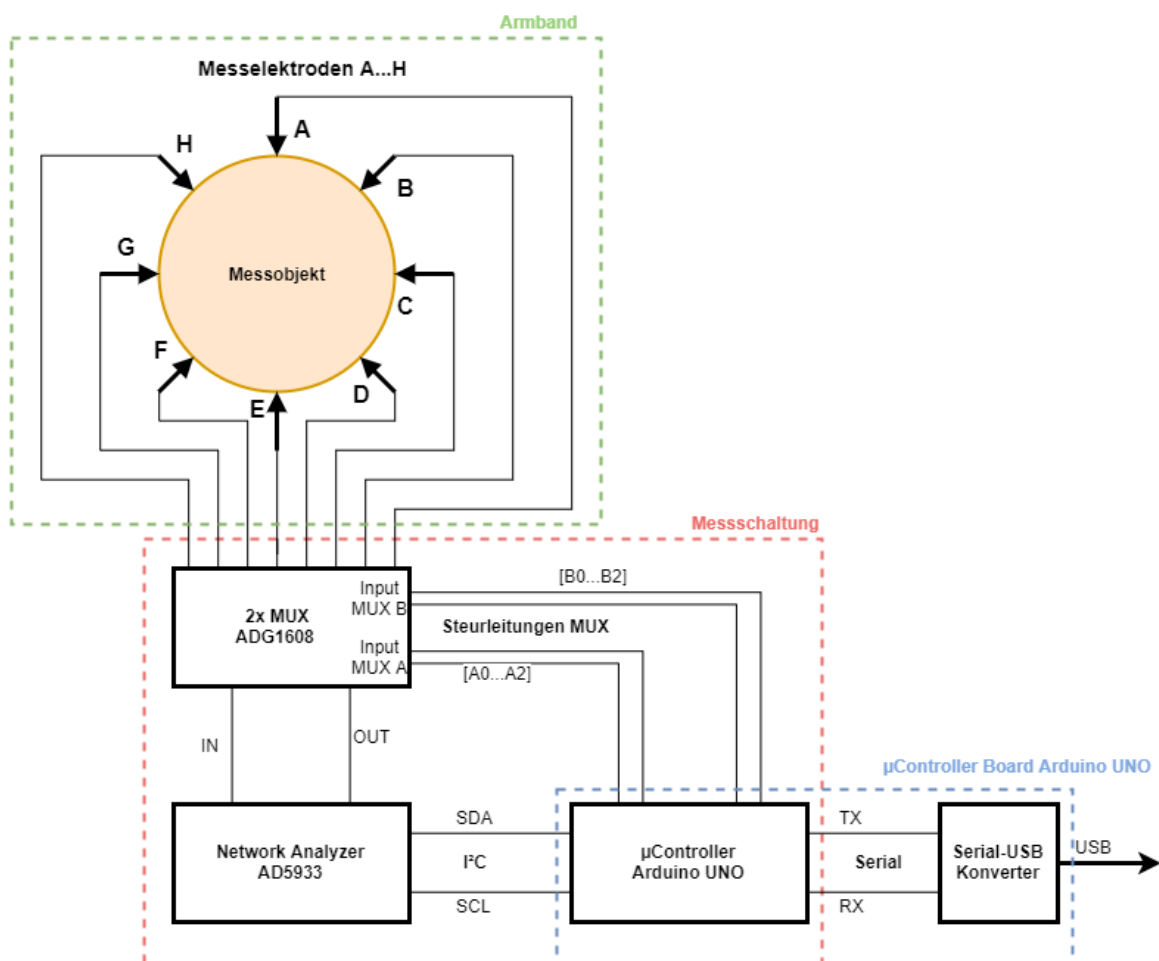


Abbildung 14: Blockschaltbild der Messeinrichtung

3.1 Messplatine

Auf der Messplatine befinden sich alle Bauteile die benötigt werden, um einen Messdurchgang mit 28 Messwerten zu bekommen. Die Platine wird über den Arduino Uno versorgt (siehe Abbildung 15). Der Aufbau der Messschaltung wird in den folgenden Unterkapitel erläutert.

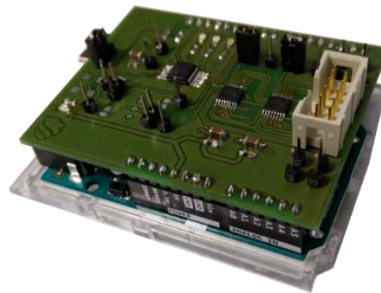


Abbildung 15: Fertige Messplatine mit dem Arduino Uno

3.1.1 AD5933

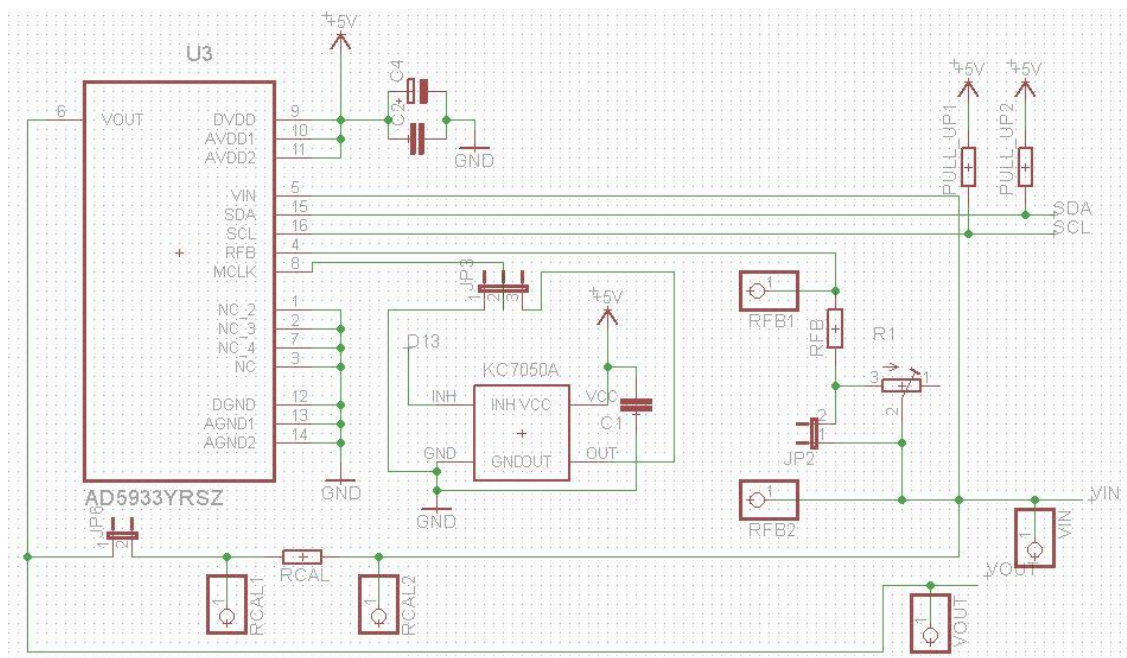


Abbildung 16: AD5933 im Schaltplan

Der AD5933 wird mit einer Versorgungsspannung von 5V versorgt und mittels Stützkondensatoren gegen Ground stabilisiert. Für den I²C Bus werden jeweils für SDA und SCL Leitungen ein Pull-Up Widerstand verwendet. Falls das erzeugte Signal des Networkanalyzers vom erwarteten Messsignal stark abweicht, wurde ein

Die Leitungen VIN und VOUT werden weiter zu den Multiplexern geleitet, um die 28 Kombinationen nacheinander messen zu können (siehe Abbildung 16).

The diagram illustrates a 16-channel signal conditioner circuit. It consists of two ADG1608BRUZ multiplexers, labeled U1 and U2, and an 8x8 switch matrix labeled COM1.

U1 Multiplexer:

- Inputs:** D0 (VIN), D1 (D6), D2 (D7), D3 (D8), D4 (D9).
- Control:** EN (D9), VSS (GND), VDD (+5V).
- Outputs:** S1, S2, S3, S4, S5, S6, S7, S8.

U2 Multiplexer:

- Inputs:** D0 (VOUT1), D1 (D2), D2 (D3), D3 (D4), D4 (D5).
- Control:** EN (D5), VSS (GND), VDD (+5V).
- Outputs:** S1, S2, S3, S4, S5, S6, S7, S8.

COM1 Switch Matrix:

- Inputs:** 1, 2, 3, 4, 5, 6, 7, 8.
- Outputs:** 1, 2, 3, 4, 5, 6, 7, 8.

Output Channels:

- VOUT1, VOUT2, VOUT3, VOUT4, VOUT5, VOUT6, VOUT7, VOUT8.

The circuit is powered by a +5V supply, with decoupling capacitors C7 and C8 for U1, and C5 and C6 for U2.

- 15 -

Bei den beiden Multiplexern handelt es sich um den IC ADG1608 [11]. Dieser Baustein besitzt je acht Kanäle. Zu den acht Elektroden wurden je ein Kanal vom Multiplexer zugewiesen (siehe Abbildung 17). Bei einem Multiplexer wird das Signal VOUT eingespeist und beim anderen Multiplexer wird VIN weitergeleitet. Das Umschalten zwischen den Kanälen erfolgt über eine 3Bit Adressleitung (siehe Tabelle 2). Die Adressleitungen sind mit digitalen I/O Pins am Arduino Uno verbunden (siehe Abbildung 17 & 19).

A0	A1	A2	Ausgang
0	0	0	S1
0	0	1	S2
0	1	0	S3
0	1	1	S4
1	0	0	S5
1	0	1	S6
1	1	0	S7
1	1	1	S8

Tabelle 2: Umschaltmöglichkeiten der Multiplexer

Damit der Networkanalyzer ohne die Multiplexer getestet werden kann, wurde vor den Multiplexer Jumper eingebaut. Beim Versorgen der Multiplexer ist es nötig die VSS Eingänge gegen Ground (GND) verbinden, weil keine negative Spannungen benötigt wird. Bei jedem Kanal befindet sich je ein Pin Header, damit das Signal leichter mit dem Oszilloskop gemessen werden kann, sodass die Funktionstüchtigkeit der Multiplexer getestet werden können.

3.1.3 Status-LEDs

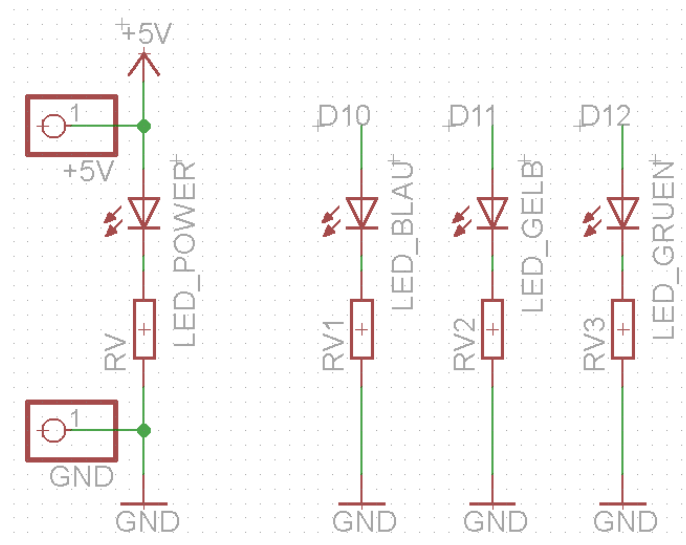


Abbildung 18: Alle LEDs auf dem Schaltplan

Es befinden sich auf der Platine vier LEDs (siehe Abbildung 18) die dem Benutzer helfen zu erkennen, in welchem Status sich der Mikrokontroller gerade befindet. Mit 480Ω Vorwiderständen wird der Strom für die LEDs begrenzt.

LED_POWER: Die LED leuchtet, wenn die Messplatine versorgt wird.

LED_BLAU: Die LED leuchtet, wenn Messwerte an den PC gesendet werden.

LED_GELB: Die LED leuchtet, wenn die Messung gestartet hat.

LED_GRUEN: Die LED leuchtet, wenn die Initialisierung erfolgreich durchgeführt worden ist.

3.2 Phantom

Ein Phantom wird in der Medizin verwendet um einen Körper bzw. Körperteile nachzubilden, an dem Messungen durchgeführt werden. Deshalb wird in diesem Projekt zum Bemessen von Objekten ein Phantomaufbau benutzt.

Als Phantom wurde eine runde Kunststoffschale verwendet, als Nachbild eines Unterarmquerschnitts. Auf der Schale wurden acht Elektroden in gleichen Abständen von $22,5^\circ$ geklebt. Als Elektroden wurden selbstklebende Kupferfolien verwendet. Auf die Folien wurden Litzendrähte gelötet, damit die Flexibilität nicht

beeinträchtigt wird. An dem anderen Ende wurden die Drähte an ein Breakout Board gelötet, damit die Messplatine und das Phantom einfach verbinden lassen.

3.3 Arduino Uno

Um mit allen IC's zu kommunizieren war es nötig sich für einen Mikrokontroller zu entscheiden. Für den Prototypen wurde als Master das Arduino Uno System [13] in Erwägung gezogen, da das Board alle erforderlichen Eigenschaften für das Projekt erfüllt. Der Arduino Uno wurde mit dem AD5399 über die serielle Schnittstelle I²C verbunden. Die Multiplexer Steuereingänge werden jeweils an einen digitalen Eingang des Arduinos angeschlossen (siehe Abbildung 16 und 17). Des Weiteren steuert der Arduino Uno die drei LEDs auf der Messplatine.

Die Aufgaben des Arduino Unos sind:

- Messung des AD5399 starten
- Messwerte vom AD5399 lesen
- Kanäle im Multiplexer umschalten
- Messwerte an den PC senden

Damit die Messplatine auf das Arduino Uno Board passt, wurde die Schaltung dementsprechend angepasst wie in Abbildung 19 ersichtlich.

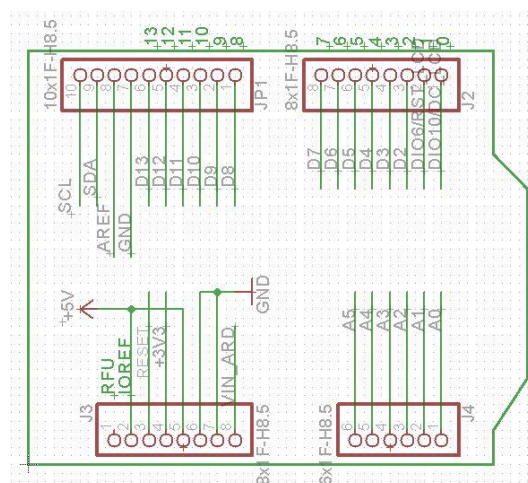


Abbildung 19: Arduino Layout im Schaltplan

3.3.1 I²C Schnittstelle

Am Anfang wurde die I²C Übertragung überprüft, ob der Takt stabil ist und Daten gesendet werden. Die Leitungen wurden mit einem Oszilloskop gemessen. In der Abbildung 20 kann beobachtet werden, dass der Takt als auch die Datenübertragung einer Rechteckssignal ähneln aber nicht ideal sind. Dafür sind die 5,8k Ω Pull Up Widerstände verantwortlich, denn diese wurden etwas zu hoch dimensioniert. Bei der Dimensionierung wurde nicht auf die Buskapazität der I²C Leitung geachtet. Die Kapazität beträgt 400pF [10]. Mit den gewählten Widerstandswerten ist die Zeitkonstante $\tau = R \cdot C$ hoch. Damit ein sauberes Rechteckssignal entsteht, kann statt dem 5,8k Ω ein 4,7k Ω Widerstand verwendet werden. Trotzdem war die Datenübertragung erfolgreich und es wurden keine Probleme bei der Funktionalität festgestellt. In der Abbildung 20 ist auch ersichtlich, dass die Taktfrequenz in Standard Mode bei 100kHz betrieben wird.

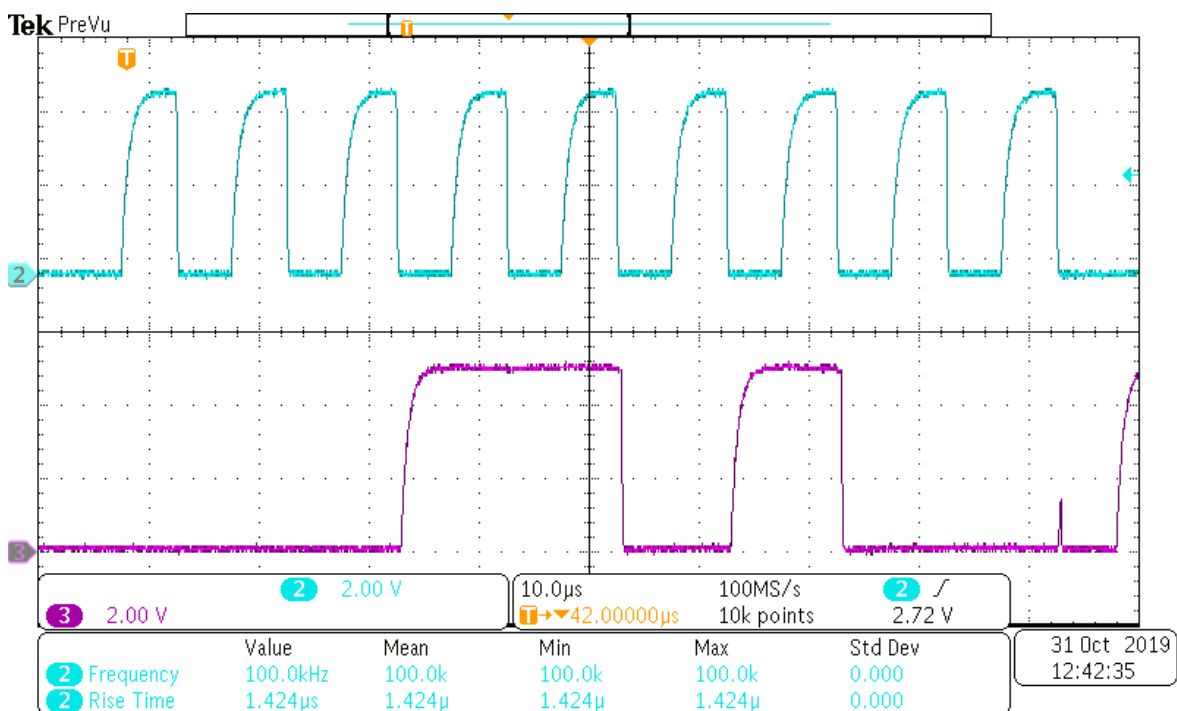


Abbildung 20: Oszilloskop Bild von der I²C Übertragung. X-Achse: 10µs/dev Y-Achse: 2V/dev

3.3.2 VOUT Signal

Die Register von dem Networkanalyzer wurde beschrieben, sodass der interne Oszillator ein 40kHz Signal erzeugt (siehe Programmcode im Anhang). Mit einem Oszilloskop wurde an dem VOUT Pin Header das Signal untersucht. In der

Abbildung 21 kann erkannt werden, dass das eingestellte Signal um 1,81kHz abweicht und ein reines Sinussignal erzeugt wird. Also ist der AD5933 für eine Messung funktionstüchtig. Die Abweichung hat keinen Einfluss auf die Messdurchführung.

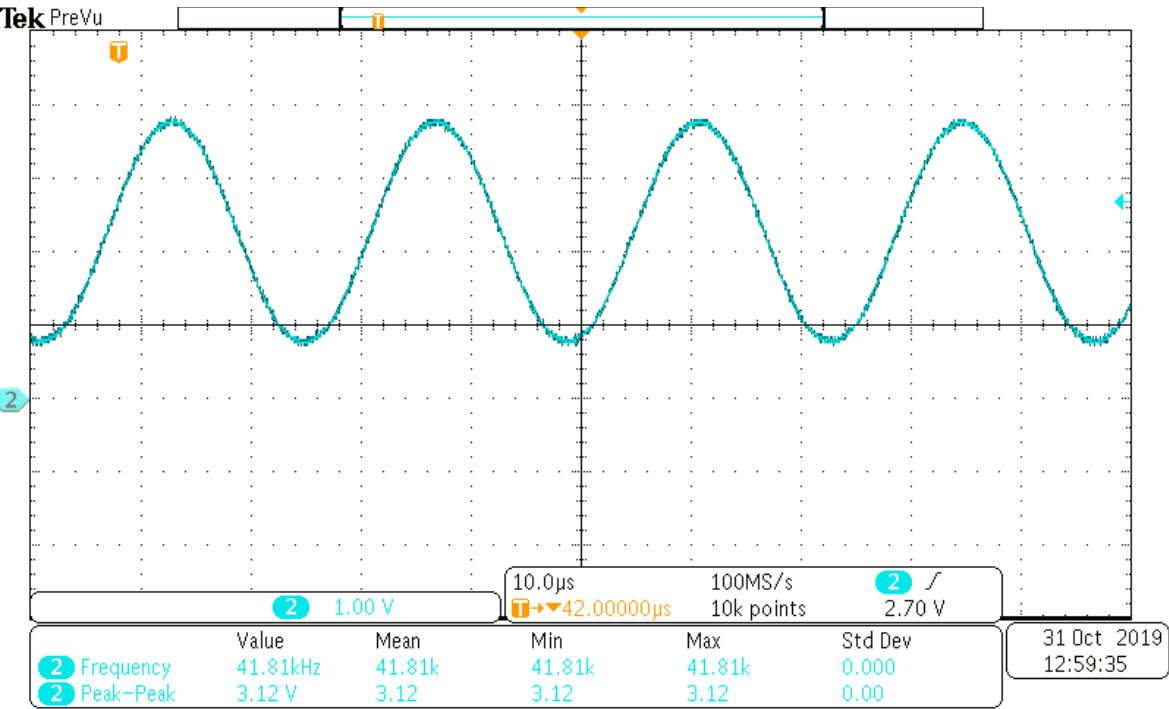


Abbildung 21: Oszilloskop Bild vom VOUT. X-Achse: 10µs/dev Y-Achse: 1V/dev

4 Programmcode

In diesem Kapitel wird das Flussdiagramm des Programmcodes näher erläutert (siehe Abbildung 22), der vollständige Programmcode befindet sich im Anhang.

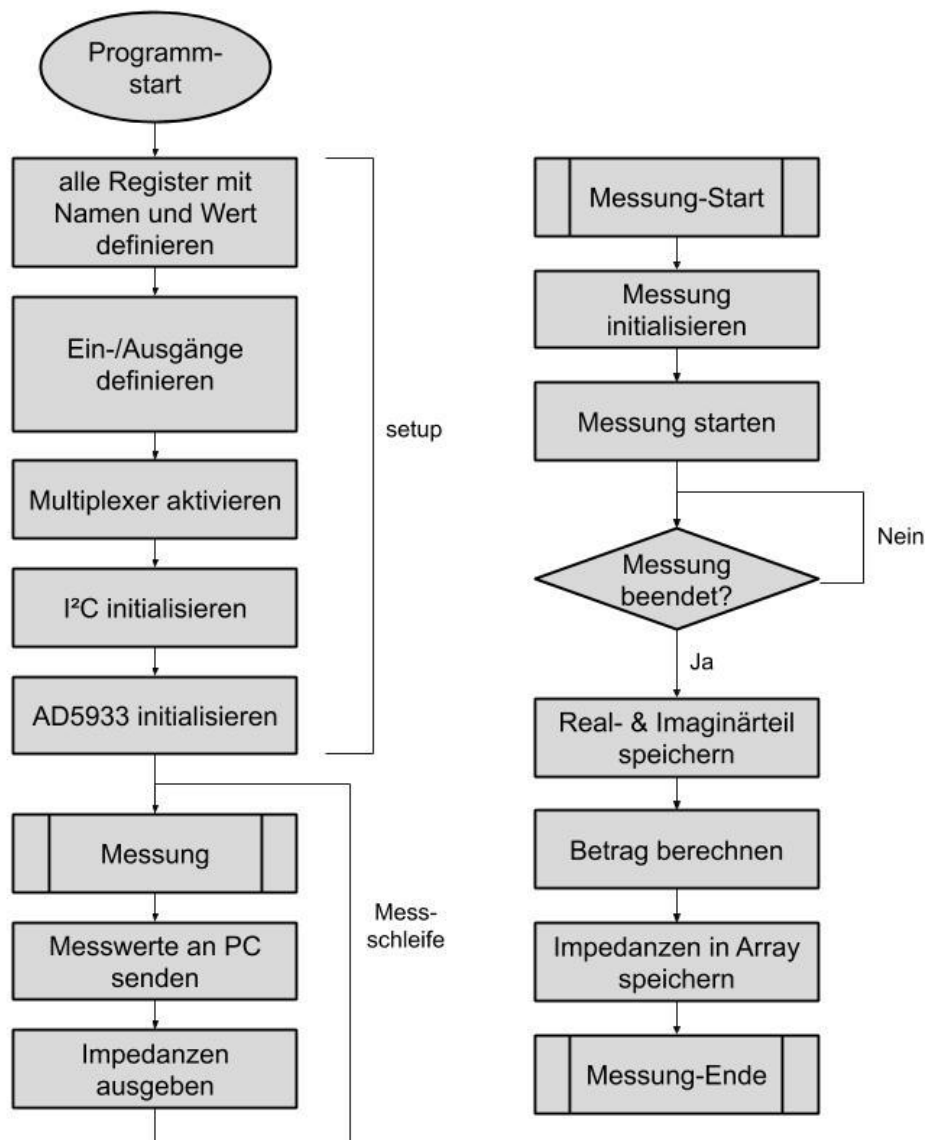


Abbildung 22: Flussdiagramm vom Code

Das Programm startet am Anfang mit der Initialisierung und wird einmalig aufgerufen. Der Bereich „setup“ im Flussdiagramm stellt die Initialisierung dar. In diesem werden alle Register mit Namen und Werten versehen, alle Ein- und Ausgänge definiert, der Multiplexer eingeschaltet, die I²C Schnittstelle und der Networkanalyzer initialisiert. Der nächste große Programmblock ist die „Messung“.

Die „Messung“ befindet sich in einer Schleife und führt die einzelnen Messungen zwischen den Elektroden durch und die Messwerte werden in einem Array abgespeichert. Das Unterprogramm „Messung“ wird 28 mal durchgeführt und anschließend werden die 28 Messwerte über die Serielle Schnittstelle an den Computer übertragen.

An dem Computer konnten die Werte mit der Software vom Arduino abgelesen werden. Hiernach wurden die Messwerte, beschrieben in [12], über die serielle Schnittstelle mit MATLAB erfasst, um später die Bildbearbeitung durchführen zu können.

5 Messung und Auswertung

In diesem Abschnitt der Arbeit wird die Durchführung der Messungen erläutert. Anschließend werden die Messwerte ausgewertet.

5.1 Messaufbau

Um eine Messung durchzuführen müssen alle Teile zusammengesetzt werden. Die Messplatine wird mit eines Flachbandkabels mit dem Breakout Board verbunden (siehe Abbildung 23). Mittels Phantom können nun Objekte in einem gewünschten Layout platziert werden. Auf den Arduino Uno Board wird die Messplatine gesetzt. Der Arduino Uno wird über ein USB Kabel mit dem PC verbunden, um nach der Messung die Messwerte zu senden.

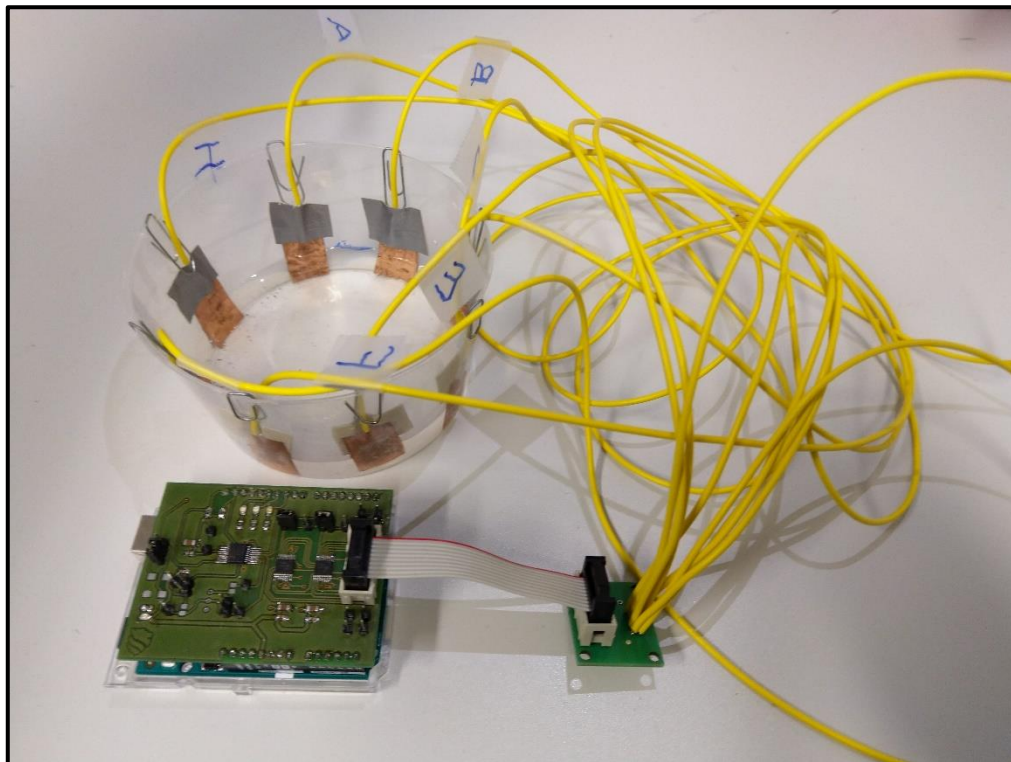


Abbildung 23: Gesamtaufbau des Projekts

5.2 Kalibration der Messung

Um Widerstandsänderungen zu erfassen muss das Messsystem kalibriert werden. Dazu muss der RFB mit Hilfe des RCAL kalibriert werden. Dabei wurden verschiedene Widerstandswerte mit einem externen Widerstandsnetzwerk

ausprobiert. Bei diesem Aufbau wurde das Phantom mit Wasser bis zu der Markierung aufgefüllt (siehe Abbildung 24). Danach wurden Kunststoffobjekte oder Metallobjekte, wie in Abbildung 24 zu sehen ist, platziert. Nach Tests an Messobjekten hat sich herausgestellt, dass bei einem Wert von $1\text{k}\Omega$ die Messwerte sich genügend ändern um Widerstandsänderungen festzustellen. Somit wurde $1\text{k}\Omega$ als RFB eingelötet.

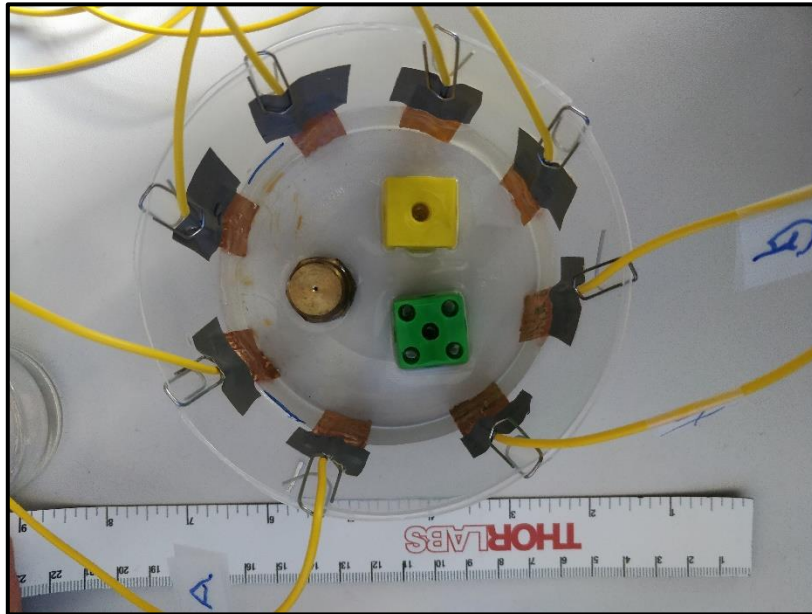


Abbildung 24: Phantom mit Messobjekten

5.3 Betrag und Gain Factor Berechnungen

Der Mikrokontroller liest aus dem Networkanalyzer binäre Werte. Diese Werte entsprechen nicht den realen Widerstandswerten. Aufgrund dessen wird der Betrag mit den im Kapitel 2.4 erwähnten Real- und Imaginärteil von DFT berechnet. Der Betrag wird weiters mit einem Gain Factor in den realen Widerstandswert skaliert.

5.3.1 Betrag

Um die Impedanz nach einer Messung auszuwerten, wird ihr Betrag benötigt. Aus dem Networkanalyzer werden die gespeicherten Werte vom Realteil und Imaginärteil von den jeweiligen Register ausgelesen.

- Registeradresse des Realteils **R**: 0x94h MSB & 0x95h LSB
- Registeradresse des Imaginärteils **I**: 0x96h MSB & 0x97h LSB

Der ausgelesene Wert hat eine Größe von 8 Bit. Damit man den gesamten Wert vom Realteil oder Imaginärteil bekommt, werden jeweils zwei Registerwerte zusammengefasst. Der Wertebereich liegt zwischen 0 und 2^{16} . Mit den beiden zusammengefassten Werten lässt sich der Betrag mit der Formel $Betrag = \sqrt{R^2 + I^2}$ berechnen.

5.3.2 Gain Factor

Der Gain Factor hat die Aufgabe, den errechneten Betrag zu skalieren um die wahre Impedanz zu erhalten. Um den Gain Factor zu ermitteln, wird zunächst ein Widerstand bemessen, dessen Wert gleich dem RFB entspricht. Durch die Bemessung der gleich großen Widerstände bekommt man einen Betrag im Mittelpunkt des Messbereichs. Der Betrag von $1k\Omega$ beträgt 9097. Somit lässt sich der Gain Factor mit der Formel

$$Gain\ Factor = \frac{\left(\frac{1}{Impedanz}\right)}{Betrag} = \frac{\left(\frac{1}{1k\Omega}\right)}{9097\Omega} = 1,09926e - 7 \frac{1}{\Omega^2}$$

berechnen.

5.3.3 Impedanz

Für die weiteren Messungen bleibt der Gain Factor gleich, solange die Messwerte sich im linearen Bereich aufhalten. Als Folge dessen ändert sich der Betrag je nach dem wie die Objekte im Phantom platziert sind. Somit lässt sich mit Leichtigkeit die Impedanz mit folgender Formel errechnen.

$$Impedanz = \frac{1}{Gain\ Factor \cdot Betrag} = \frac{1}{1,09926e - 7 \cdot Betrag}$$

Weil im Microcontroller mit ganzen Zahlen gerechnet werden, wird die Berechnung

$$\frac{1}{Gain\ Factor} = \frac{1}{1,09926e - 7} = 9097000 \Omega^2$$

durchgeführt und in den Code als

$$Impedanz [\Omega] = \frac{9097000 [\Omega^2]}{Betrag [\Omega]}$$

eingetragen.

5.4 Das Verhalten der 28 Messwerte

Am Phantom sind acht Elektroden befestigt, damit sind in einem Messdurchgang 28 Messungen durchführbar. Die Elektroden wurden mit den Buchstaben von A bis H für einen besseren Überblick markiert (siehe Abbildung 25). Eine Messung wird mit zwei Elektroden durchgeführt und man erhält einen Messwert für die Strecke zwischen den Elektroden. Bei einem Messdurchgang nimmt die Anzahl an Messstrecken pro Elektrode ab, somit werden die Strecken nicht doppelt bemessen.

Anzahl der Messungen pro Elektrode:

- Elektrode A: 7
 - Elektrode B: 6
 - Elektrode C: 5
 - Elektrode D: 4
 - Elektrode E: 3
 - Elektrode F: 2
 - Elektrode G: 1
- Total: 28 Messungen

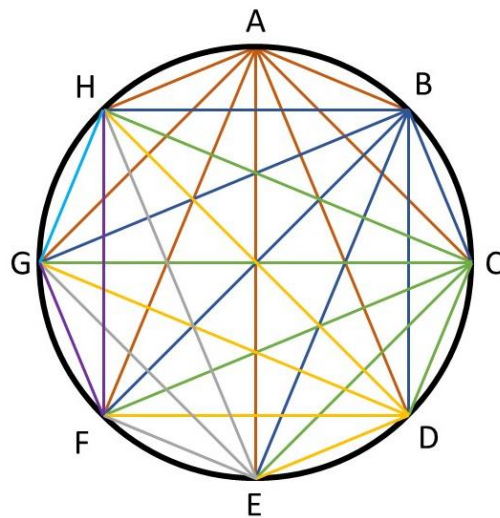


Abbildung 25: Alle 28 Messmöglichkeiten

5.4.1 Messung ohne Objekt

Beim ersten Durchgang befindet sich im Phantom Wasser und es wurden keine Objekte platziert, um zu prüfen ob die Messwerte sich richtig verhalten (siehe Abbildung 26 und Tabelle 3).

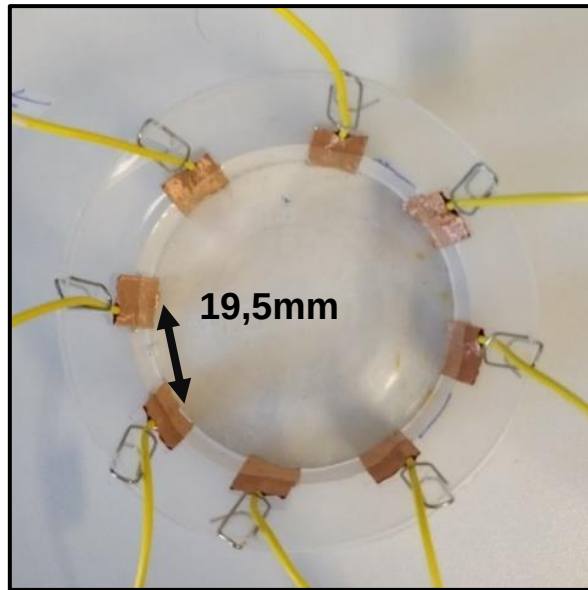


Abbildung 26: Phantom ohne Messobjekte mit Wasser aufgefüllt

Messpunkte	Ω	Messpunkte	Ω
AB	1470	CE	2000
AC	2030	CF	2220
AD	2290	CG	2160
AE	2325	CH	2240
AF	2270	DE	1300
AG	2010	DF	2000
AH	1640	DG	2050
BC	1555	DH	2300
BD	2050	EF	1475
BE	2225	EG	1810
BF	2300	EH	2140
BG	2150	FG	1200
BH	2005	FH	1900
CD	1550	GH	1510

Tabelle 3: 28 Widerstandswerte des Messaufbaus

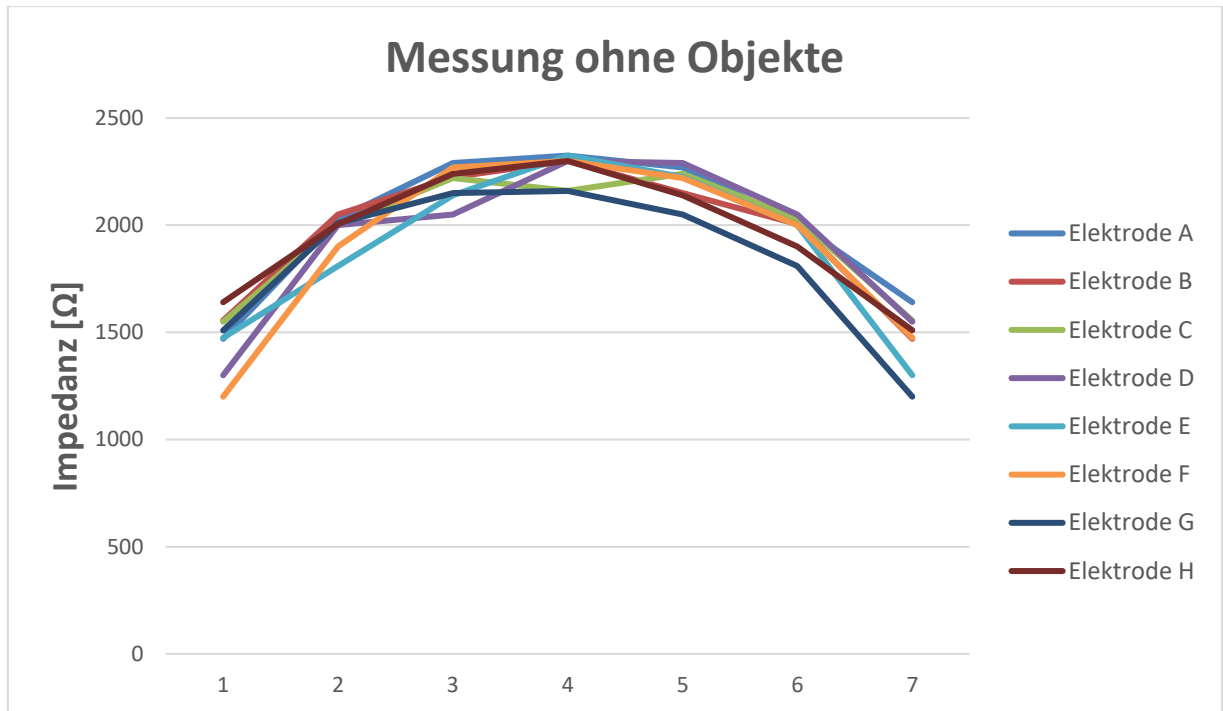


Diagramm 1: Messwertverhalten ohne Testobjekte

An dem Diagramm 1 kann erkannt werden, dass mit steigender Distanz auch der Widerstandswert steigt. Die Messwerte verhalten sich wie erwartet. Mit den Widerständen lässt sich der spezifische Widerstand vom Wasser berechnen, dazu wird die folgende Formel benötigt:

$$\text{spezifischer Widerstand} \left[\frac{\Omega \text{mm}^2}{\text{m}} \right] \rho = \text{Widerstand} [\Omega] R \cdot \frac{\text{Fläche} [\text{mm}^2] A}{\text{Länge} [\text{m}] l}$$

Für die Berechnung wird aus Tabelle 3 der Widerstandswert zwischen der Strecke AB verwendet. Die Kontaktfläche der Elektroden, die mit Wasser in Berührung steht, beträgt 225mm² (gemessen 15mm mal 15mm). Die Länge zwischen den Elektroden A und E beträgt 0,0195m.

$$\rho = 1470\Omega \cdot \frac{225\text{mm}^2}{0,0195\text{m}} = 1,6962 \cdot 10^7 \frac{\Omega \text{mm}^2}{\text{m}}$$

Das Ergebnis der Berechnung ist tolerabel, denn der spezifische Widerstand von Leitungswasser beträgt $2 \cdot 10^7 \frac{\Omega \text{mm}^2}{\text{m}}$ [14].

5.4.2 Messung mit einem Kunststoffkapsel

Im zweiten Durchlauf wurde eine Plastikkapsel in das Phantom platziert (siehe Abbildung 27).

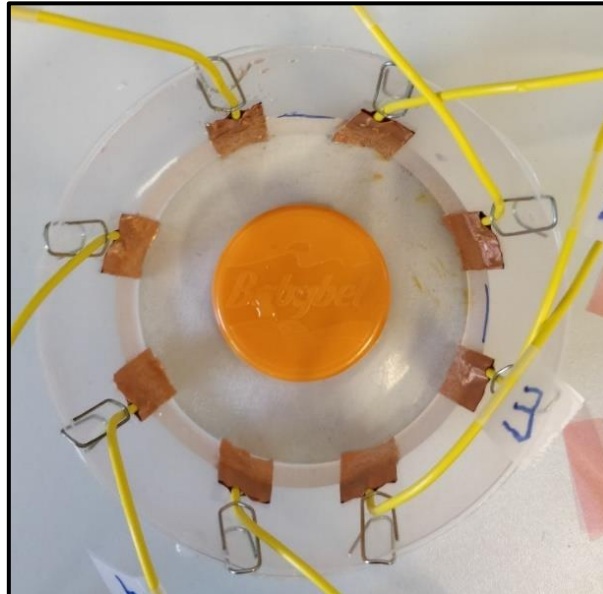


Abbildung 27: Messaufbau mit einem Kunststoffkapsel

Messpunkte	Ω	Messpunkte	Ω
AB	1210	CF	2025
AC	1850	CG	2125
AD	2160	CH	2140
AE	2210	DE	1170
AF	2150	DF	1640
AG	1830	DG	1950
AH	1348	DH	2170
BC	1275	EF	1040
BD	1810	EG	1450
BE	2075	EH	1850
BF	2175	FG	1090
BG	2075	FH	1650
BH	1825	GH	1150
CD	1225		
CE	1750		

Tabelle 4: 28 Widerstandswerte vom Messaufbau mit einem Kunststoffkapsel

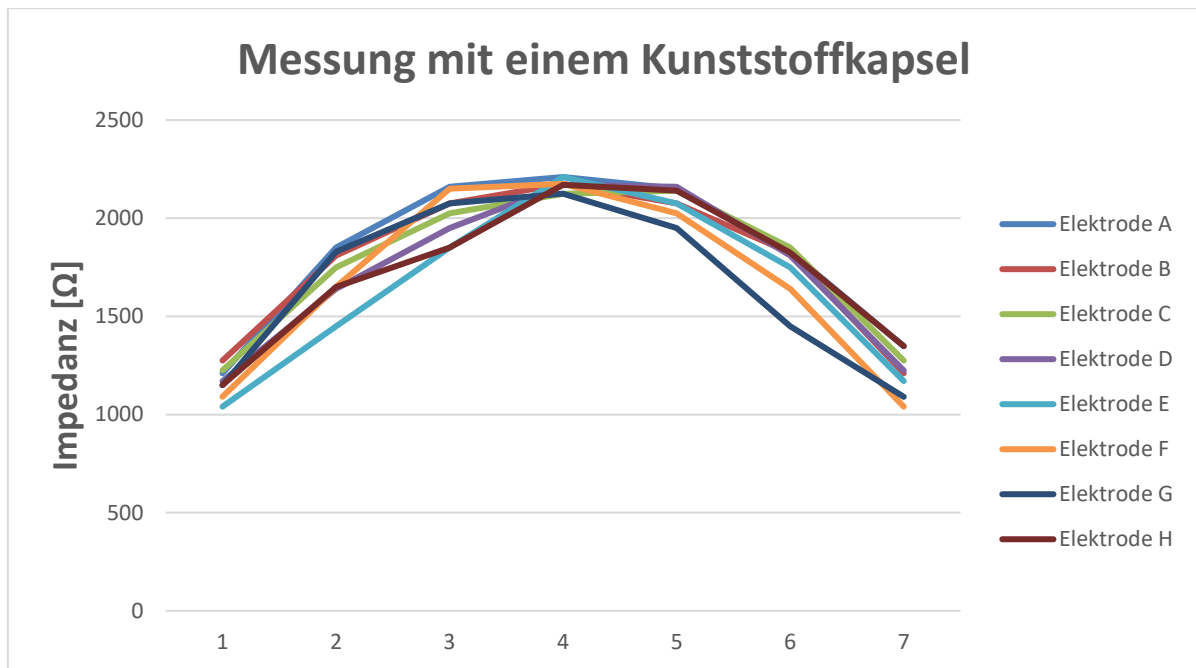


Diagramm 2: Messwertverhalten bei einem Kunststoffkapsel

An dem Diagramm 2 wird nichts außergewöhnliches beobachtet, wie erwartet zeigt die Kunststoffkapsel dielektrische Eigenschaften. Bei den Messwerten wiederum kann gesehen werden, dass diese von der ersten Messung ohne Objekte abweicht. Die Tabelle 4 verglichen mit Tabelle 3 zeigt, dass die Werte bei Tabelle 4 niedriger sind als bei Tabelle 3. Die Ursache für dieses Verhalten ist die Höhe vom Wasserspiegel. Als die Kapsel im Wasser platziert war, stieg somit der Wasserspiegel an und die Elektroden bekamen mehr Kontaktfläche. Durch die größere Kontaktfläche fließt der Strom leichter und der Widerstand nimmt ab.

5.4.3 Messung mit Kunststoff und Metall Objekten

Bei der letzten Messung wurden auch metallische Objekte in die Messung einbezogen, um das Verhalten zu untersuchen (siehe Abbildung 28).

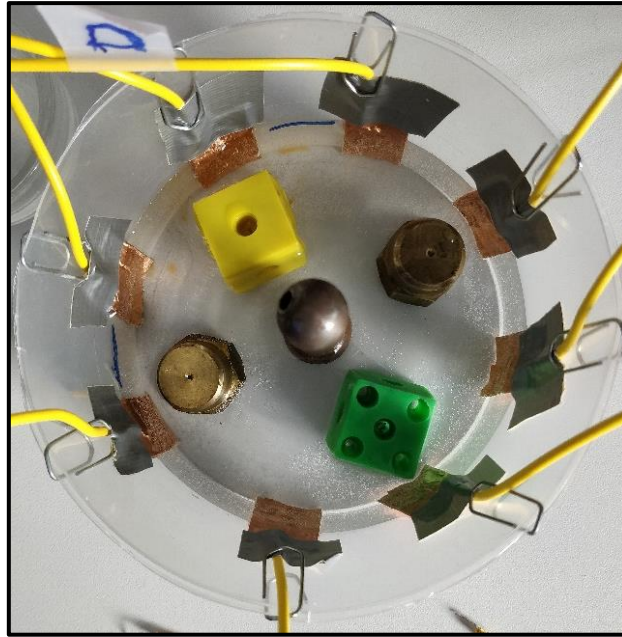


Abbildung 28: Messaufbau mit Kunststoffwürfeln und Metallobjekte

Messpunkte	Ω	Messpunkte	Ω
AB	1010	CE	1550
AC	1610	CF	1850
AD	1610	CG	2050
AE	1460	CH	1925
AF	1610	DE	1040
AG	1575	DF	1460
AH	1200	DG	1825
BC	1375	DH	1800
BD	1600	EF	1050
BE	1550	EG	1550
BF	1725	EH	1600
BG	1790	FG	1270
BH	1500	FH	1600
CD	1325	GH	1170

Tabelle 5: 28 Widerstandswerte vom Messaufbau mit Kunststoffwürfeln und Metallobjekte

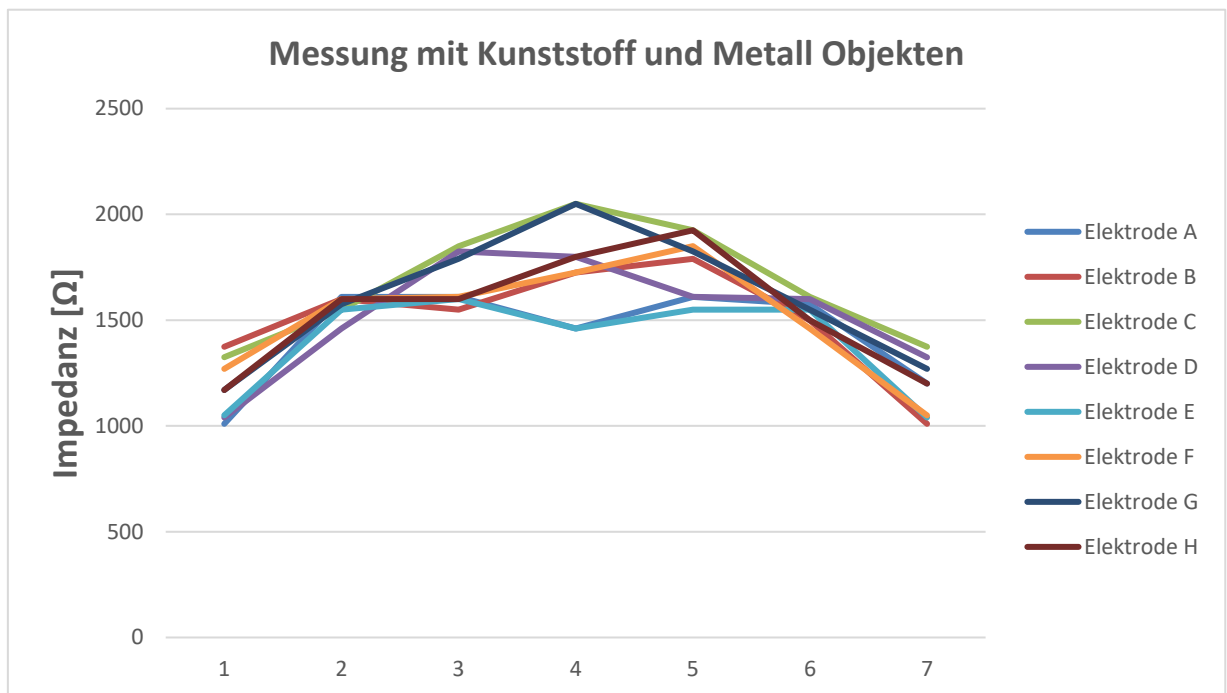


Diagramm 3: Messwertverhalten von der Abbildung 24.

An dem Diagramm 3 kann erkannt werden, dass die Widerstandswerte (siehe Tabelle 5) an den Strecken niedriger sind, auf dem sich die metallischen Objekte befinden. Zugleich ist ersichtlich, dass an den Positionen mit den Kunststoffwürfeln der Widerstand deutlich höher ist z.B. an der Strecke von den Elektroden C und G. Mit dieser Erkenntnis kann die Darstellung der Bilder in dem zweiten Teil der Arbeit [12] erstellt werden.

5.5 Vergleich der Kennlinien

Damit die Widerstandsänderung besser sichtbar wird, wurden zwei Elektroden verglichen. Die Elektrode A und C wurden für das Vergleich gewählt, da sich bei der dritten Messung auf der Strecke AE metallische Objekte befinden und auf der Strecke CG ein metallisches Objekt und zwei Kunststoffobjekte.

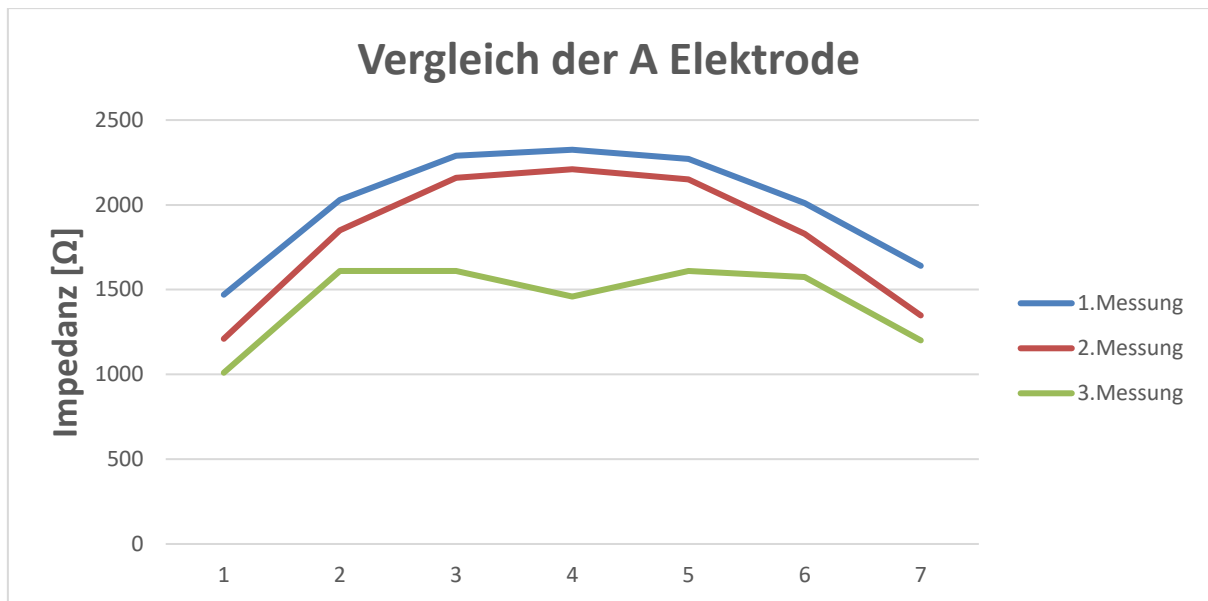


Diagramm 4: Messverhalten der drei Messungen von der Elektrode A

Das Diagramm 4 zeigt, dass die Messwerte zwischen erster und zweiter Messung sich ähnlich verhalten, da bei 2. Messung ein nicht leitfähiges Objekt sich befindet. Im Gegensatz zu der 3. Messung ist es zu erkennen, dass an der vierten Stelle (A und E) der Widerstand sich verringert.

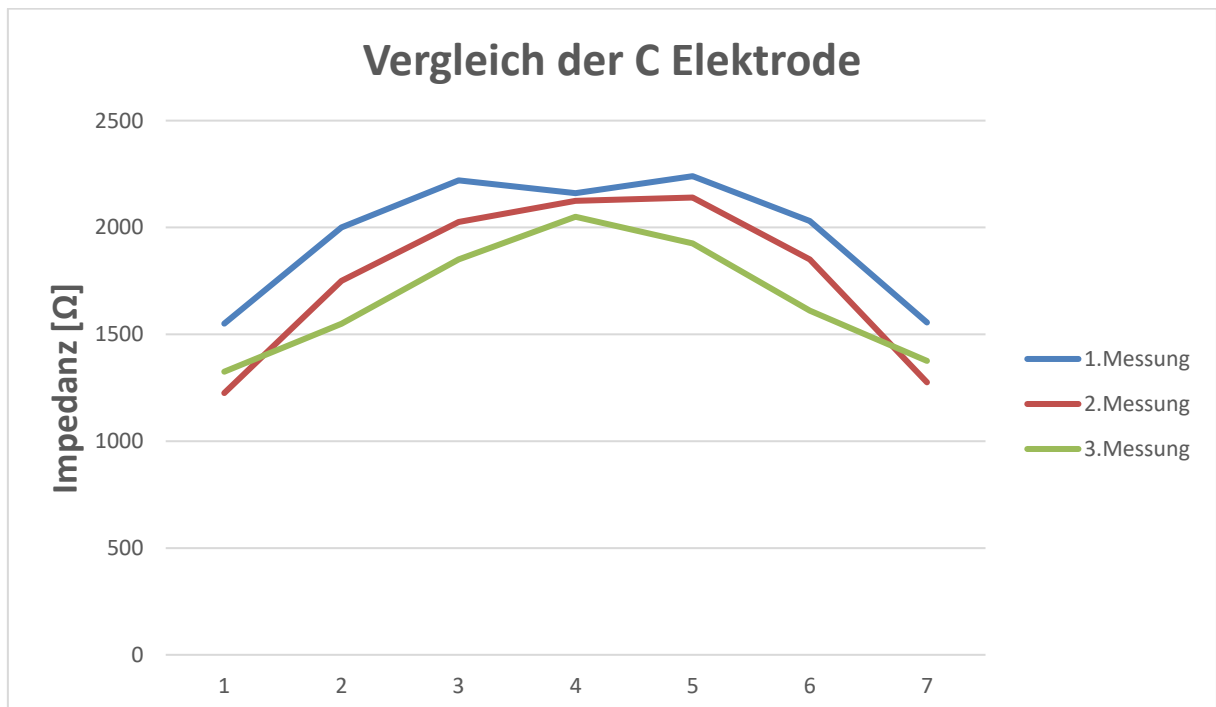


Diagramm 5: Messverhalten der drei Messungen von der Elektrode C

An dem Diagramm 5 werden die Widerstandsveränderungen der C Elektrode etwas näher betrachtet. Bei der 3. Messung an der vierten Strecke (C und G) befindet sich ein höherer Widerstand als auf dem Diagramm 4. Der hohe Widerstand auf der Strecke entsteht wegen den Kunststoffobjekten, diese wirken isolierend und somit wird der Widerstand erhöht. Die 1. Messung weist an der vierten Stelle auf einen Ausreißer, der nicht stark von den restlichen Werten abweicht. Sonst verhalten sich die Messwerte korrekt.

6 Conclusio & Outlook

6.1 Conclusio

Es wurde ein Prototyp für Impedanzmessung entwickelt und mit Hilfe eines Testaufbaus Messungen durchgeführt. Das entworfene und implementierte System basiert auf dem Networkanalyzer AD5933.

Nach einem Kalibrierungsprozess kann die Messplatine zum Messen verwendet werden. Bei unserem Aufbau hat sich für die sichtbare Messwertänderung $1\text{k}\Omega$ als Kalibrierungswiderstand passend erwiesen. Bei einem Messdurchgang werden 28 Werte an dem Testaufbau gemessen und wurden in einem Diagramm dargestellt. Je nach Platzierung der Objekte wurden unterschiedliche örtliche Widerstandsverteilungen aufgenommen.

Die erste Messung zeigt (siehe Diagramm 1), dass sich der Widerstand mit steigender Distanz erhöht. Bei der zweiten Messung konnten keine Veränderungen festgestellt werden, weil die Kunststoffkapsel nicht leitet. Die dritte Messung wiederum zeigt, dass der Strom bei den Strecken zwischen metallischen Gegenständen weniger Widerstand hat.

Die Messwerte von der Messplatine wurden zur Weiterverarbeitung mittels USB Kabel an den PC übertragen. Die Weiterverarbeitung wird in der Bachelorarbeit Matthias Bernhart beschrieben. In seinem Teil wurde die Bildbearbeitung durchgeführt. Die erfassten Bilder haben gezeigt, dass die 8 Elektroden ausreichend sind um Widerstandsveränderungen zu sehen, aber nicht um kleine Strukturen zu orten [12].

Obwohl die Geräteleistung einige Einschränkungen aufwies, wurden alle festgelegten Ziele erreicht. Die angewendeten Prinzipien sind vielversprechend und ermöglichen die Erfassung großer Datenmengen, die für den Informationsabruf statistisch behandelt werden können. Mit einigen Hardwareverbesserungen könnte ein sehr attraktives System für Gestenerkennung geschaffen werden.

6.2 Outlook

Der nächste Schritt dieses Projektes wäre die Schaltung neu zu entwerfen. Die neue Schaltung sollte aus einer Platine bestehen, auf dem sich der Mikrokontroller, der Networkanalyzer und der Elektrodenstecker befinden. Die Schaltung sollte zusätzlich mit einem drahtlosem Modul erweitert werden, damit die Datenübertragung an den PC für den Nutzer angenehmer wird.

Des Weiteren ist es nötig ein Armband zu entwickeln, in dem sich die Elektroden leicht einbauen lassen. Der Armband sollte in der Größe verstellbar sein, da der Umfang vom Unterarm bei Personen variieren kann.

Abbildungsverzeichnis

Abbildung 1: Myo Armband.....	2
Abbildung 2: a) Ein Foto des tragbaren Systems mit Referenzmünze. b) Systems mit Details am Handgelenk getragen, und der gleichen Referenzmünze. [5].....	3
Abbildung 3: 16 Gesten die mit dem Armband erkennbar sind. [5].....	3
Abbildung 4: Lichtquelle mit dem Schatten. [6].....	4
Abbildung 5: Schaltzeichen der Impedanz mit der anliegenden Spannung und der durchfließenden Strom.....	5
Abbildung 6: Ohmsch-induktive Impedanz	6
Abbildung 7: Ohmsch-induktive Impedanz	6
Abbildung 8: Ersatzschaltbild der Zelle	7
Abbildung 9: I ² C Bus mit zwei Pull Up Widerständen, einem Master und zwei Slaves	8
Abbildung 10: Zeitverhalten bei der Übertragung mit dem Acknowledge Bit	8
Abbildung 11: Aufbau des I ² C Protokolls.....	9
Abbildung 12: Start- und Stoppbedingung von der Datenübertragung	9
Abbildung 13: Blockschaltbild vom AD5933 [9]	11
Abbildung 14: Blockschaltbild der Messeinrichtung.....	13
Abbildung 15: Fertige Messplatine mit dem Arduino Uno	14
Abbildung 16: AD5933 im Schaltplan.....	14
Abbildung 17: Beide Multiplexer mit den Ausgängen	15
Abbildung 18: Alle LEDs auf dem Schaltplan	17
Abbildung 19: Arduino Layout im Schaltplan.....	18
Abbildung 20: Oszilloskop Bild von der I ² C Übertragung. X-Achse: 10µs/dev Y-Achse: 2V/dev	19
Abbildung 21: Oszilloskop Bild vom VOUT. X-Achse: 10µs/dev Y-Achse: 1V/dev.....	20
Abbildung 22: Flussdiagramm vom Code.....	21
Abbildung 23: Gesamtaufbau des Projekts	23
Abbildung 24: Phantom mit Messobjekten	24

Abbildung 25: Alle 28 Messmöglichkeiten	26
Abbildung 26: Phantom ohne Messobjekte mit Wasser aufgefüllt	27
Abbildung 27: Messaufbau mit einem Kunststoffkapsel	29
Abbildung 28: Messaufbau mit Kunststoffwürfeln und Metallobjekte	31

Tabellenverzeichnis

Tabelle 1: I ² C Moden mit der Übertragungsgeschwindigkeiten	10
Tabelle 2: Umschaltmöglichkeiten der Multiplexer	16
Tabelle 3: 28 Widerstandswerte des Messaufbaus	27
Tabelle 4: 28 Widerstandswerte vom Messaufbau mit einem Kunststoffkapsel.....	29
Tabelle 5: 28 Widerstandswerte vom Messaufbau mit Kunststoffwürfeln und Metallobjekte	31

Diagrammverzeichnis

Diagramm 1: Messwertverhalten ohne Testobjekte	28
Diagramm 2: Messwertverhalten bei einem Kunststoffkapsel	30
Diagramm 3: Messwertverhalten von der Abbildung 24.	32
Diagramm 4: Messverhalten der drei Messungen von der Elektrode A	33
Diagramm 5: Messverhalten der drei Messungen von der Elektrode C	33

Literaturverzeichnis

- [1] N. Forsblom, "Were You Aware of All These Sensors In Your Smartphone?", 2015 [online] available: (15. Februar 2020) <https://blog.adtile.me/2015/11/12/were-you-aware-of-all-these-sensors-in-your-smartphone/>
- [2] Huawei Technologies, „Virtual Reality/Augmented Reality White Paper“, CAICT, 2017 [online] available: (16. Februar 2020) <https://www-file.huawei.com/-/media/CORPORATE/PDF/ilab/vr-ar-en.pdf/>
- [3] Y. Zhang & C. Harrison, „Tomo: Wearable, Low-Cost, Electrical Impedance Tomography for Hand Gesture Recognition“, ACM, Carnegie Mellon University, Pittsburgh, 2015. [online] available: (05. Jänner 2020) <http://www.chrisharrison.net/projects/tomo/tomo.pdf>
- [4] Y. Ploengpit and T. Phientrakul, "Rock-paper-scissors with Myo Armband pose detection," *2016 International Computer Science and Engineering Conference (ICSEC)*, Chiang Mai, 2016. [online] available: (05. Jänner 2020) <https://ieeexplore-ieee-org.wn.idm.oclc.org/document/7859949>
- [5] A. Ferrone, X. Jiang, L. Maiolo, A. Pecora, L. Colace and C. Menon, "A fabric-based wearable band for hand gesture recognition based on filament strain sensors: A preliminary investigation," *2016 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT)*, Cancun, 2016. [online] available: (05. Jänner 2020) <https://ieeexplore-ieee-org.wn.idm.oclc.org/document/7797710>
- [6] C. Lin and L. Deng, "Shadow approach for real-time hand gesture recognition using a light-dependent resistor array," *2017 IEEE Sensors Applications Symposium (SAS)*, Glassboro, NJ, 2017. [online] available: (05. Jänner 2020) <https://ieeexplore-ieee-org.wn.idm.oclc.org/document/7894066>
- [7] F. Deimel & A. Hasenzagl, „Grundlagen der Elektrotechnik Band 2“, *Veritas Verlag GmbH*, ISBN 978-3-7058-7602-6, 11. Auflage, Linz, 2011.
- [8] B. Kainka, „Erfolgreich Messen, Steuern und Regeln mit Mikrocontrollern: serielle Interface-Schaltungen in der Praxis“, *Franzis Verlag GmbH*, ISBN 3-7723-4723-1, 2. Auflage, Poing, 1998.

- [9] Analog Devices, „AD5933“, U.S.A. [online] available: (05. Jänner 2020) <https://www.analog.com/media/en/technical-documentation/data-sheets/AD5933.pdf>
- [10] Atmel, „ATmega328P“, U.S.A. [online] available: (23. Jänner 2020) http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- [11] Analog Devices, „ADG1608/ADG1609“, U.S.A. [online] available: (23. Jänner 2020) https://www.analog.com/media/en/technical-documentation/data-sheets/ADG1608_1609.pdf
- [12] Matthias Bernhart, „Rekonstruktion von elektrischen Impedanz-Tomographieschnittbilder“, Fachhochschule Wiener Neustadt, Wiener Neustadt, 2020.
- [13] Arduino [online] available: (16. Februar 2020) <https://www.arduino.cc>
- [14] Wikipedia, Spezifischer Widerstand, zuletzt geändert 10. Dezember 2019 [online] available: https://de.wikipedia.org/wiki/Spezifischer_Widerstand

Anhang

Verwendete Software

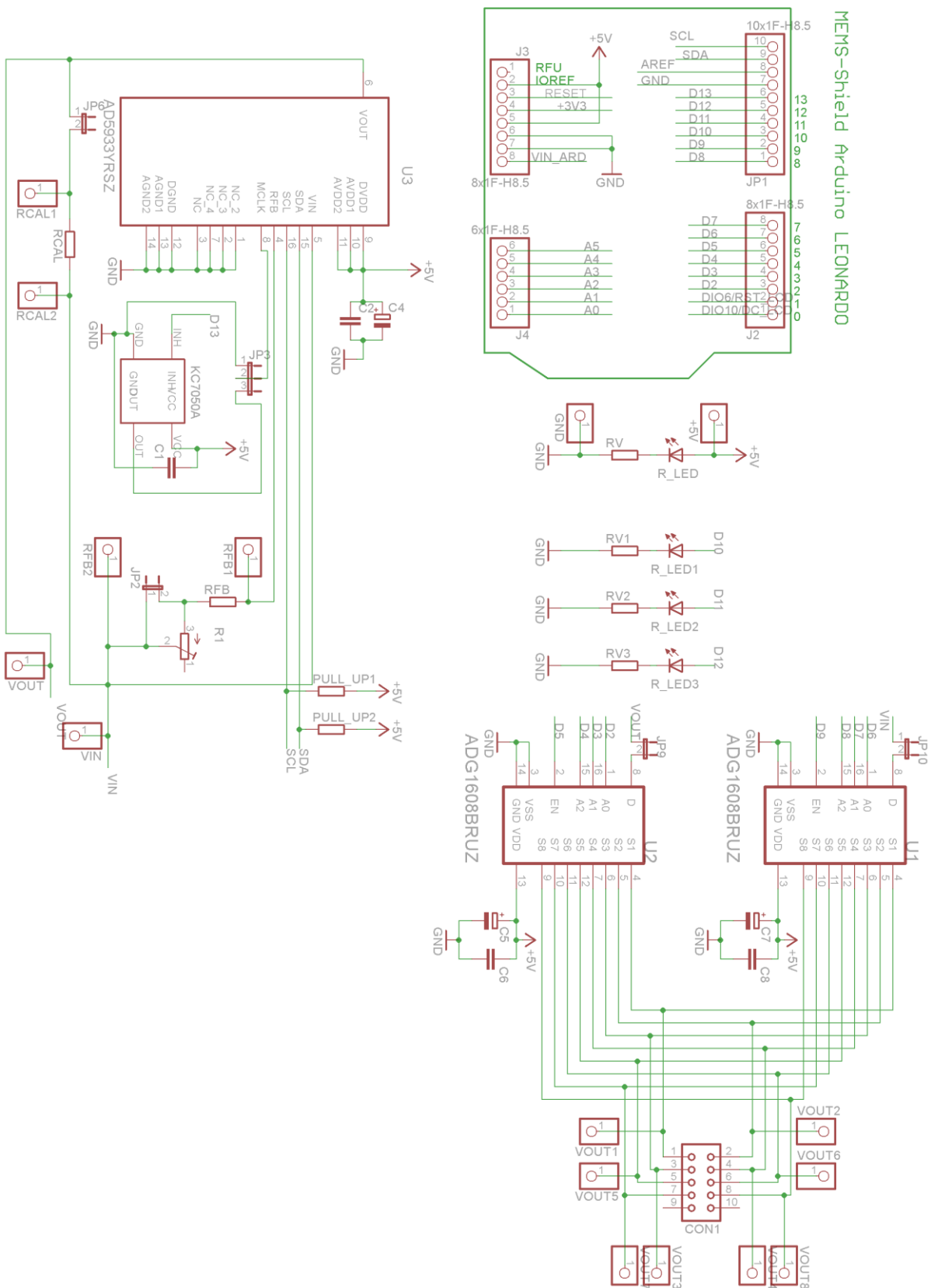
- **Eagle**

Eagle ist ein Electronic Design Automation Programm. In diesem Programm wurden die Schaltplane gezeichnet und die Layouts erstellt.

- **Arduino IDE**

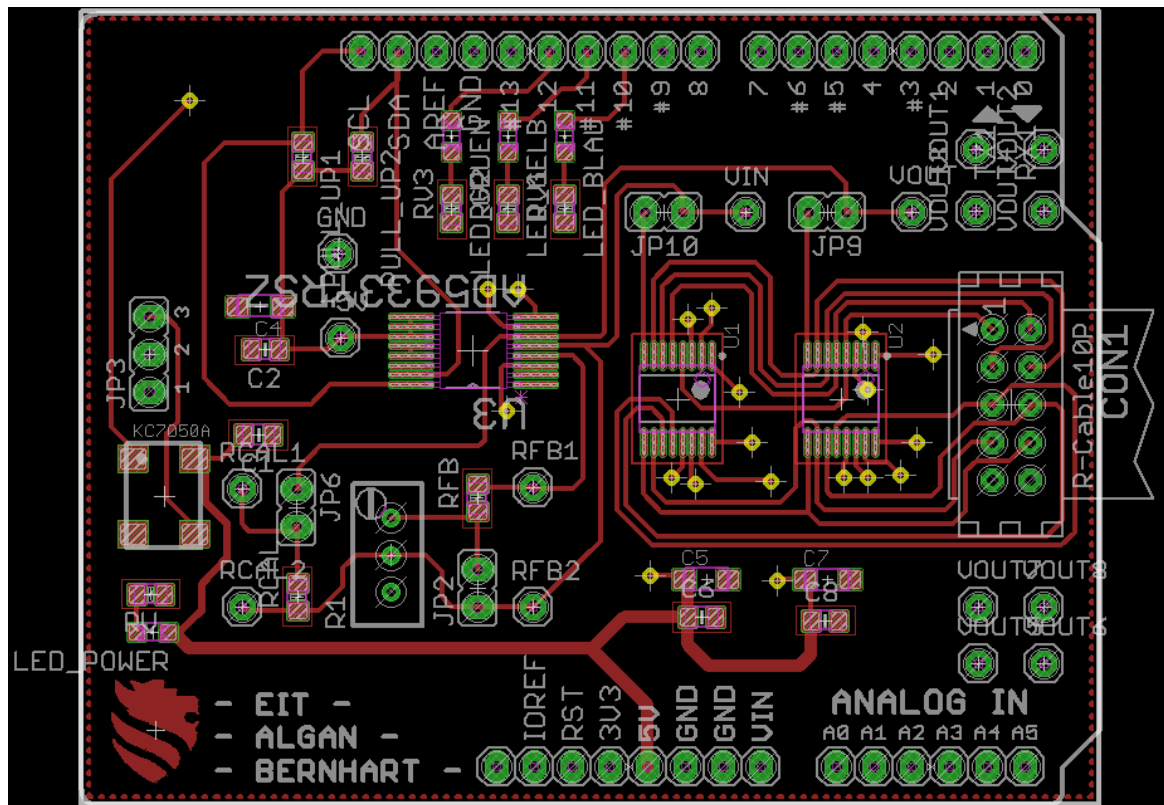
Der Code in für den Mikrokontroller wurde in Arduino IDE geschrieben. Dieses Software ist von Arduino selbst erstellt worden, um das Arbeiten mit den Arduino Boards zu vereinfachen.

Gesamt Schaltplan



[illegible]

Top Layout



Programmcode

```
#include <Wire.h>
#include <math.h>
#define A_0 2
#define A_1 3
#define A_2 4
#define en_A 5
#define B_0 6
#define B_1 7
#define B_2 8
#define en_B 9

const char AD5933_addr = 0x0D; // Default Slave Address
const char Addr_Pointer = 0xB0; // Pointer Address
/*****

Register-Map

*****/

const char Control_H_addr = 0x80;
const char Control_L_addr = 0x81;
const char Start_Freq_H_addr = 0x82;
```

```

const char Start_Freq_M_addr = 0x83;
const char Start_Freq_L_addr = 0x84;
const char Freq_Inc_H_addr = 0x85;
const char Freq_Inc_M_addr = 0x86;
const char Freq_Inc_L_addr = 0x87;
const char Num_of_Inc_H_addr = 0x88;
const char Num_of_Inc_L_addr = 0x89;
const char Num_of_Settl_t_cycles_H_addr = 0x8A;
const char Num_of_Settl_t_cycles_L_addr = 0x8B;
const char Status_addr = 0x8F;
const char T_Date_H_addr = 0x92;
const char T_Date_L_addr = 0x93;
const char R_Date_H_addr = 0x94;
const char R_Date_L_addr = 0x95;
const char I_Date_H_addr = 0x96;
const char I_Date_L_addr = 0x97;

```

```

/*****

```

Variable

```

*****/

```

```

// various frequencies for START FREQUENCY REGISTER

```

```

const char S_Freq_30_H = 0x3D;      // D23-D16
const char S_Freq_30_M = 0x70;      // D15-D8
const char S_Freq_30_L = 0xA3;      // D7-D0
//const char S_Freq_30_H = 0x0F;    // D23-D16
//const char S_Freq_30_M = 0x5C;    // D15-D8
//const char S_Freq_30_L = 0x28;    // D7-D0
const char S_Freq_40_H = 0x14;
const char S_Freq_40_M = 0x7A;
const char S_Freq_40_L = 0xE1;
const char S_Freq_50_H = 0x19;
const char S_Freq_50_M = 0x99;
const char S_Freq_50_L = 0x99;

```

```

const char S_Freq_60_H = 0x1E;
const char S_Freq_60_M = 0xB8;
const char S_Freq_60_L = 0x51;

// FREQUENCY INCREMENT REGISTER not used
const char Freq_Inc_H = 0x00;      // D23-D16
const char Freq_Inc_M = 0x00;      // D15-D8
const char Freq_Inc_L = 0x00;      // D7-D0

// NUMBER OF INCREMENTS REGISTER not used
const char Num_of_Inc_H = 0x00;     // D15-D8
const char Num_of_Inc_L = 0x01;     // D7-D0

// NUMBER OF SETTling TIME CYCLES REGISTER
const char Num_of_cycles_H = 0x00;  // D15-D8
const char Num_of_cycles_L = 0xFF;  // D7-D0

// Variables for read/write
char Control_H_init = 0x11;         // D15-D8
char Control_H_standby = 0xB1;
char Control_H_start = 0x21;
char Control_L = 0x00;              // D7-D0

char Status;                        // D7-D0
char T_Data_H = 0;                  // D15-D8
char T_Data_L = 0;                  // D7-D0
char R_Data_H = 0;                  // D15-D8
char R_Data_L = 0;                  // D7-D0
char I_Data_H = 0;                  // D15-D8
char I_Data_L = 0;                  // D7-D0

signed int measure_R, measure_L;
unsigned int magn;

```

```

int z[28];
int i = 1;

void WW(char x, char y) // Method for Write Command
{
    Wire.beginTransmission(AD5933_addr);
    Wire.write(x);
    Wire.write(y);
    //Wire.endTransmission();
    if (Wire.endTransmission() != 0)
    {
        Serial.println("Error-write");
    }
}

char RR(char z) // Method for Read Command
{
    char c;
    Wire.beginTransmission(AD5933_addr);
    Wire.write(Addr_Pointer);
    Wire.write(z);
    //Wire.endTransmission();
    if (Wire.endTransmission() != 0)
    {
        Serial.println("Error-read");
    }

    Wire.requestFrom(AD5933_addr, 1, true);
    while (Wire.available())
    {
        c = Wire.read();
    }
    return c;
}

```

```

void init_AD9533() // Method for initializing the IC AD5933
{
    WW(Start_Freq_H_addr, S_Freq_40_H);    // selecting the used frequency
    WW(Start_Freq_M_addr, S_Freq_40_M);
    WW(Start_Freq_L_addr, S_Freq_40_L);

    WW(Num_of_Inc_H_addr, Num_of_Inc_H); // selecting the increment, in this
    WW(Num_of_Inc_L_addr, Num_of_Inc_L); // case zero

    WW(Freq_Inc_H_addr, Freq_Inc_H); // defines how much frequency each
    WW(Freq_Inc_M_addr, Freq_Inc_M); // increment rises, in this case zero
    WW(Freq_Inc_L_addr, Freq_Inc_L);

    WW(Num_of_Settl_t_cycles_H_addr, Num_of_cycles_H);
    WW(Num_of_Settl_t_cycles_L_addr, Num_of_cycles_L);

    WW(Control_H_addr, Control_H_standby); // stand by mode
    WW(Control_L_addr, Control_L);
}

float measure_AD5933() // Methode for measure
{
    WW(Control_H_addr, Control_H_init); // initalazing the AD5933 before measure
    WW(Control_L_addr, Control_L);
    delay(100);
    WW(Control_H_addr, Control_H_start); // starting the measurement
    WW(Control_L_addr, Control_L);

    while (!(RR(Status_addr) & 0x02)) // wait until measument is done
    {
        //Serial.println("waitin");
    }
    R_Data_H = RR(R_Date_H_addr); // storing the data
    R_Data_L = RR(R_Date_L_addr);
}

```

```

I_Data_H = RR(I_Date_H_addr);
I_Data_L = RR(I_Date_L_addr);

WW(Control_H_addr, Control_H_standby); // back to stand by mode
WW(Control_L_addr, ControMel_L);

measure_R = ((R_Data_H) << 8) + (R_Data_L); // shifting 8 bit data into 16 bit
measure_L = ((I_Data_H) << 8) + (I_Data_L);
magn = sqrt(pow(measure_R, 2) + pow(measure_L, 2)); // calculating magnitude

return (9097000/(magn)); // returing the impedance
}

void transmit(int arr[]) // method for sending the data tot he PC
{
    int n, i;
    char data[56];
    char START1 = 0xFF, START2 = 0;
    for (i = 0; i < 28; i++)
    {
        data[2 * i] = (arr[i] >> 8);
        data[2 * i + 1] = arr[i];
    }
    for (n = 0; n < 58; n++)
    {
        if (n == 0)
        {
            Serial.print(START1);
        }
        else if (n == 1)
        {
            Serial.print(START2);
        }
    }
}

```

```

    else
    {
        Serial.print(data[n - 2]);
    }
}
}

void setup() // runs once
{
    pinMode(A_0, OUTPUT);      // in-/output
    pinMode(A_1, OUTPUT);
    pinMode(A_2, OUTPUT);
    pinMode(en_A, OUTPUT);
    pinMode(B_0, OUTPUT);
    pinMode(B_1, OUTPUT);
    pinMode(B_2, OUTPUT);
    pinMode(en_B, OUTPUT);

    digitalWrite(en_A, HIGH);  // Mux enabled
    digitalWrite(en_B, HIGH);

    Wire.begin();             // Initializing I2C
    Serial.begin(19200)

    delay(2000);
    Serial.println("LOS:");
    Serial.println("");
    init_AD9533();
}

void loop()
{
    int n, m;

```

```
// Starting from here measuring all 28 values
```

```
//AB
```

```
digitalWrite(A_0, LOW);
```

```
digitalWrite(A_1, LOW);
```

```
digitalWrite(A_2, LOW);
```

```
digitalWrite(B_0, HIGH);
```

```
digitalWrite(B_1, LOW);
```

```
digitalWrite(B_2, LOW);
```

```
z[0]=measure_AD5933();
```

```
//AC
```

```
digitalWrite(A_0, LOW);
```

```
digitalWrite(A_1, LOW);
```

```
digitalWrite(A_2, LOW);
```

```
digitalWrite(B_0, LOW);
```

```
digitalWrite(B_1, HIGH);
```

```
digitalWrite(B_2, LOW);
```

```
z[1]=measure_AD5933();
```

```
//AD
```

```
digitalWrite(A_0, LOW);
```

```
digitalWrite(A_1, LOW);
```

```
digitalWrite(A_2, LOW);
```

```
digitalWrite(B_0, HIGH);
```

```
digitalWrite(B_1, HIGH);
```

```
digitalWrite(B_2, LOW);
```

```
z[2]=measure_AD5933();
```

```
//AE
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[3]=measure_AD5933();

//AF
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[4]=measure_AD5933();

//AG
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[5]=measure_AD5933();
```

```
//AH
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[6]=measure_AD5933();

//BC
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, LOW);

z[7]=measure_AD5933();

//BD
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, LOW);

z[8]=measure_AD5933();
```

```
//BE
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[9]=measure_AD5933();

//BF
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[10]=measure_AD5933();

//BG
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[11]=measure_AD5933();
```

```
//BH
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[12]=measure_AD5933();

//CD
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, LOW);

z[13]=measure_AD5933();

//CE
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[14]=measure_AD5933();
```

```
//CF
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[15]=measure_AD5933();
```

```
//CG
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[16]=measure_AD5933();
```

```
//CH
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[17]=measure_AD5933();
```

```
//DE
digitalWrite(A_0, HIGH);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[18]=measure_AD5933();

//DF
digitalWrite(A_0, HIGH);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[19]=measure_AD5933();

//DG
digitalWrite(A_0, HIGH);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[20]=measure_AD5933();
```

```
//DH
digitalWrite(A_0, HIGH);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, LOW);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[21]=measure_AD5933();

//EF
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, LOW);
digitalWrite(B_2, HIGH);

z[22]=measure_AD5933();

//EG
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);
digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[23]=measure_AD5933();
```

```
//EH
digitalWrite(A_0, LOW);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[24]=measure_AD5933();
```

```
//FG
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);

digitalWrite(B_0, LOW);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[25]=measure_AD5933();
```

```
//FH
digitalWrite(A_0, HIGH);
digitalWrite(A_1, LOW);
digitalWrite(A_2, HIGH);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[26]=measure_AD5933();
```

```

//GH
digitalWrite(A_0, LOW);
digitalWrite(A_1, HIGH);
digitalWrite(A_2, HIGH);

digitalWrite(B_0, HIGH);
digitalWrite(B_1, HIGH);
digitalWrite(B_2, HIGH);

z[27]=measure_AD5933();

for(n=0;n<28;n++)
{
  Serial.print("Z:");
  Serial.println(z[n]);
  if(n==27)
  {
    Serial.println("");
  }
}

transmit(z); // sending all 28 values the PC
delay(20);
}

```

DATASHEET ab nächster Seite